

NAVAL POSTGRADUATE SCHOOL

Monterey, California



THESIS

J2125 ✓

UTILIZATION OF A KALMAN OBSERVER WITH
LARGE SPACE STRUCTURES

by

Bruce M. Jackson

December 1988

Thesis Advisor

Jeff B. Burl

Approved for public release; distribution is unlimited.

T241978

REPORT DOCUMENTATION PAGE

Report Security Classification Unclassified		1b Restrictive Markings	
Security Classification Authority		3 Distribution Availability of Report	
Declassification Downgrading Schedule		Approved for public release; distribution is unlimited.	
Performing Organization Report Number(s)		5 Monitoring Organization Report Number(s)	
Name of Performing Organization	6b Office Symbol (if applicable) 33	7a Name of Monitoring Organization	
Naval Postgraduate School		Naval Postgraduate School	
Address (city, state, and ZIP code)		7b Address (city, state, and ZIP code)	
Monterey, CA 93943-5000		Monterey, CA 93943-5000	
Name of Funding Sponsoring Organization	8b Office Symbol (if applicable)	9 Procurement Instrument Identification Number	
Address (city, state, and ZIP code)		10 Source of Funding Numbers	
		Program Element No	Project No Task No Work Unit Accession No

Title (include security classification) UTILIZATION OF A KALMAN OBSERVER WITH LARGE SPACE STRUCTURES

Personal Author(s) Bruce M. Jackson

Type of Report	13b Time Covered	14 Date of Report (year, month, day)	15 Page Count
Master's Thesis	From To	December 1988	85

Supplementary Notation The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

Cosati Codes			18 Subject Terms (continue on reverse if necessary and identify by block number)
Id	Group	Subgroup	Kalman Observer, large space structures.

Abstract (continue on reverse if necessary and identify by block number)

Control of the motions and vibrations of large space structures require the knowledge of state values that may not be available due either to inability to measure the states or, the high cost of the sensors to measure the required states. One solution is the use of an observer to estimate the states from limited sensor input.

The physical characteristics of large space structures and the environment they operate in will cause large amounts of noise in the measurements. The obvious observer for such an environment is the Kalman Filter which is specifically designed to produce optimal estimates in a noisy environment.

A straightforward application of the Kalman Filter will be examined utilizing a steady state Kalman gain matrix. The observer performance will be examined in both matched filter/plant and reduced order filter configurations.

Distribution Availability of Abstract		21 Abstract Security Classification	
Unclassified unlimited <input type="checkbox"/> same as report <input type="checkbox"/> DTIC users		Unclassified	
22a Name of Responsible Individual		22b Telephone (include Area code)	22c Office Symbol
Off B. Burl		(408) 646-2390	62BL

Approved for public release; distribution is unlimited.

Utilization of a Kalman Observer with Large Space Structures

by

Bruce M. Jackson
Lieutenant Commander, United States Navy
B.S., United States Naval Academy, 1976

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ENGINEERING SCIENCE

from the

NAVAL POSTGRADUATE SCHOOL
December 1988

ABSTRACT

Control of the motions and vibrations of large space structures require the knowledge of state values that may not be available due either to inability to measure the states or, the high cost of the sensors to measure the required states. One solution is the use of an observer to estimate the states from limited sensor input.

The physical characteristics of large space structures and the environment they operate in will cause large amounts of noise in the measurements. The obvious observer for such an environment is the Kalman Filter which is specifically designed to produce optimal estimates in a noisy environment.

A straightforward application of the Kalman Filter will be examined utilizing a steady state Kalman gain matrix. The observer performance will be examined in both matched filter/plant and reduced order filter configurations.

12/23
C.1

THESIS DISCLAIMER

The reader is cautioned that computer programs developed in this research may not have been exercised for all cases of interest. While every effort has been made, within the time available, to ensure that the programs are free of computational and logic errors, they cannot be considered validated. Any application of these programs without additional verification is at the risk of the user.

TABLE OF CONTENTS

I. INTRODUCTION	1
A. BACKGROUND	1
B. PROBLEM STATEMENT	1
C. ORGANIZATION	2
II. MATHEMATICAL MODEL	3
A. INTRODUCTION	3
B. MODAL MODEL	3
C. DISCRETE-TIME MODEL	6
III. THE OBSERVER	10
A. INTRODUCTION	10
B. KALMAN FILTER EQUATIONS	10
C. STEADY-STATE SOLUTION	12
D. OBSERVER PERFORMANCE	12
IV. SIMULATION	14
A. INTRODUCTION	14
B. PLANT AND OBSERVER DATA	14
C. SIMULATION PROGRAMS	14
D. EFFECT OF PLANT TO MEASUREMENT NOISE RATIO ON OB- SERVER PERFORMANCE	15
E. EFFECTS OF INCREASED MODES ON OBSERVER PERFORMANCE	16
F. REDUCED ORDER KALMAN OBSERVER	16
V. CONCLUSIONS AND RECOMMENDATIONS	35
A. CONCLUSIONS	35
B. RECOMENDATIONS	35
APPENDIX A. KALMAN GAIN MATRIX GENERATION PROGRAM	36

APPENDIX B. KALMAN OBSERVER AND PLANT SIMULATION	49
APPENDIX C. PROGRAM TO ESTIMATE NOISE IN KALMAN FILTER FROM UNOBSERVED MODES	62
LIST OF REFERENCES	73
INITIAL DISTRIBUTION LIST	74

LIST OF TABLES

Table 1.	CUMULATIVE UNANTICIPATED NOISE FROM UNOBSERVED MODES	18
Table 2.	UNANTICIPATED NOISE FROM UNOBSERVED MODES BY MODE	18

LIST OF FIGURES

Figure 1.	Observer Performance (J) $PN/MN = 1.0d12$	19
Figure 2.	Observer Performance (J) $PN/MN = 1.0d11$	20
Figure 3.	Observer Performance (J) $PN/MN = 1.0d10$	21
Figure 4.	Observer Performance (J) $PN/MN = 5.0d09$	22
Figure 5.	Observer Performance (J) $PN/MN = 2.5d09$	23
Figure 6.	Observer Performance (J) $PN/MN = 1.0d09$	24
Figure 7.	Mode 7 (Postion) Observer Performance versus PN/MN	25
Figure 8.	Settling Time versus PN/MN	26
Figure 9.	Observer Performance (J) 4 Modes (7 - 10)	27
Figure 10.	Observer Performance (J) 5 Modes (7 - 11)	28
Figure 11.	Observer Performance (J) 6 Modes (7 - 12)	29
Figure 12.	Observer Performance (J) 7 Modes (7 - 13)	30
Figure 13.	Observer Performance (J) 8 Modes (7 - 14)	31
Figure 14.	Observer Performance (J) 9 Modes (7 - 15)	32
Figure 15.	Observer Performance (J) 10 Modes (7 - 16)	33
Figure 16.	Settling Time versus number of Modes Observed	34

ACKNOWLEDGEMENTS

I would like to express my appreciation to the McDonnell Douglas Astronautics Company of Huntington Beach, California, for allowing the use of their dynamic model of a preliminary space station configuration. Also, I would like to thank Major William J. Preston, USMC for allowing the use of the mathematical model for the station dynamics which makes up the majority of Chapter II of this paper as well as portions of his simulation program. Finally, I would like to thank Professor J. B. Burl for his assistance and most of all his patience in the completion of this work.

This paper is dedicated to Christina Marie Jackson (USNA '10).

I. INTRODUCTION

A. BACKGROUND

The advent of large space structures poses a number of problems for the control engineer. Previously, the objects put into space could be treated as rigid bodies so that a single three axis sensor package could be used to tell the motion of all components. The large space structures will not be rigid, instead they will have considerable flexibility and multiple modes of vibration [Ref. 1: p. 51]

Control of the structure's attitude and vibrations requires knowing the motions of the components. One approach would be to heavily instrument the space structure, but weight and cost make this approach impractical. An alternative is to use a limited number of sensors to measure only certain states and to deduce the other required states by use of an observer algorithm.

This thesis will address the production of estimates of the states needed for control of the structure. The model used will be a early design study by McDonnell Douglas Astronautics for a dual keel space station. The techniques and problems of observation for this model are generic to all large space structures.

B. PROBLEM STATEMENT

Design of an observer for estimating the states of a large space structure breaks down into several steps. First, a mathematical model is developed for the system behavior over time. Modal analysis is used to form a system composed of decoupled second order differential equations. The use of decoupled equations allows a reduced order model to be generated by truncating the number of modal equations. A reduced order model will have all of the same mathematical qualities (and problems) but reduces the amount of time and computer resources required to do simulation.

Second, the observer is designed. The observer is designed to obtain a minimum variance estimate of the desired state values from the measurements.

Third, the observer is simulated to verify performance. Simulation runs of both a matched observer/plant system and a reduced order observer are employed. That is, the system is run where the observer is used to estimate all of the plant states and run where there are more plant states than the observer estimates.

Fourth, results are analysed and conclusions drawn based on these results. Recommendations for further areas of research are suggested based on the results and conclusions.

C. ORGANIZATION

The model of the space station is developed in Chapter II. The modal model was developed using modal analysis and discretized to form the discrete-time state equations. The data for this model was from an early design study by McDonnell Douglas Astronautics Company for a dual-keel space station. The observer and its equations are developed in Chapter III. Chapter IV is the simulation runs of the observer versus the plant. Chapter V presents conclusions and recommendations for further research.

II. MATHEMATICAL MODEL

A. INTRODUCTION

Prior to the proposed space station almost all of the objects put into space could be treated as simple rigid bodies for the purpose of mathematical modelling of their motions. The design constraints imposed by the high cost of lifting mass to orbit dictates a light, open structure with considerable flexing. Large space structures such as the space station, therefore, cannot be treated as rigid bodies. The structure is in fact lightly damped with multiple natural frequencies. The result is a structure that will vibrate for considerable periods of time whenever external forces are applied.

The space station structure can be modeled as an n-DOF (degree of freedom) system consisting of n masses, springs, and dashpots [Ref. 2: p. 173-176]. This straight forward modelling of the coupled masses produces a system of unworkable complexity. As a result, the system will be modelled in terms of the structures natural modes of vibration. The resulting system, while still complex, is at least workable.

The model will be developed in two steps. The first will be to generate the continuous-time model of the natural modes. The second will yield the discrete-time model, developed from the first model, for use in the simulation.

B. MODAL MODEL

The space station structure can be modeled as a system of discrete masses coupled by springs and dashpots. The major mechanism of damping in the structure is structural damping, the internal dissipation of energy within the members, as the structure vibrates. Structural damping can be shown to be equivalent to viscous damping and this equivalency is used in the model [Ref. 2: p. 72-73].

The energy dissipated by structural damping is:

$$W_d = \alpha X^2 \quad (1)$$

W_d = energy dissipated by structural damping

α = constant (force/displacement)

X = displacement

The energy dissipated by viscous damping is:

$$W_v = \pi c \omega X^2 \quad (2)$$

We can equate the two

$$\pi C_{eq} \omega X^2 = \alpha X^2 \quad (3)$$

yielding an equivalent viscous damping coefficient:

$$C_{eq} = \frac{\alpha}{\pi \omega} \quad (4)$$

The second order differential equation for a single viscously damped mass is:

$$m\ddot{x} + c\dot{x} + kx = F(t) \quad (5)$$

Substituting C_{eq} for c

$$m\ddot{x} + \frac{\alpha}{\pi \omega} \dot{x} + kx = F(t) \quad (6)$$

For multiple mass systems C_{eq} becomes $\frac{d}{\omega_f} \mathbf{K}$ where ω_f is the natural frequency of vibration.

The displacement of masses can be represented by the second order matrix differential equation [Ref. 3: p. 3-9],

$$\mathbf{M}\ddot{\mathbf{q}}(t) + \frac{d}{\omega_f} \mathbf{K}\dot{\mathbf{q}}(t) + \mathbf{K}\mathbf{q}(t) = \mathbf{F}(t) \quad (7)$$

\mathbf{q} = coordinate vector

\mathbf{M} = system mass matrix (diagonal)

$\frac{d}{\omega_f} \mathbf{K}$ = equivalent damping

d = damping coefficient

ω_f = frequency of oscilation of the system

\mathbf{K} = symmetric system stiffness matrix

$\mathbf{F}(t)$ = system forcing function

The above equation represents a system of second order differential equations coupled through the stiffness matrix. Decoupling can be done by expressing \mathbf{q} in terms of natural modes of vibration. The process is called modal analysis. The independent differential equations can then be treated individually. The modal equations are derived below.

First, the undamped, homogeneous form of Eq. (7)

$$\mathbf{M}\ddot{q}(t) + \mathbf{K}q(t) = 0 \quad (8)$$

is solved. Let

$$q(t) = Ax \sin(\omega t + \Theta) \quad (9)$$

$$\dot{q}(t) = Ax\omega \cos(\omega t + \Theta) \quad (10)$$

$$\ddot{q}(t) = -Ax\omega^2 \sin(\omega t + \Theta) \quad (11)$$

substituting Eq. (9) and Eq. (10) into Eq. (11)

$$[-\omega^2\mathbf{M} + \mathbf{K}]\mathbf{A}x \sin(\omega t + \Theta) = 0 \quad (12)$$

This equation has a non-trivial solution for all time if and only if:

$$[\mathbf{K} - \omega^2\mathbf{M}]x = 0 \quad (13)$$

Equation (12) has n combinations of x (natural mode shapes) and ω (natural frequencies) as solutions. These can be grouped into matrices:

$$\mathbf{X} = [x_1 x_2 \dots x_n]^T \quad (14)$$

$$\Omega^2 = \text{diag}[\omega_{o1}^2 \omega_{o2}^2 \dots \omega_{on}^2] \quad (15)$$

which satisfy the equation:

$$\mathbf{K}\mathbf{X} = \Omega^2\mathbf{M}\mathbf{X} \quad (16)$$

Several useful relations can be derived from Eq. (16). Premultiplying Eq. (16) by \mathbf{X}^T ,

$$\mathbf{X}^T\mathbf{K}\mathbf{X} = \Omega^2\mathbf{X}^T\mathbf{M}\mathbf{X} \quad (17)$$

The eigenvectors can be normalized

$$\mathbf{X}^T\mathbf{M}\mathbf{X} = \mathbf{I} \quad (18)$$

which yields

$$\mathbf{X}^T\mathbf{K}\mathbf{X} = \Omega^2 \quad (19)$$

The equations of motion can be uncoupled through the linear transformation of the coordinate system

$$q(t) = \sum_{i=1}^n x_i \eta_i(t) = \mathbf{X} \eta(t) \quad (20)$$

\mathbf{X} = modal matrix

n = maximum number of degrees of freedom

$\eta(t)$ = transformed coordinate vector

Application of the transformation to the system Eq. (7) yields

$$\mathbf{X}^T \mathbf{M} \mathbf{X} \ddot{\eta}(t) + \frac{d}{\omega_f} \mathbf{X}^T \mathbf{K} \mathbf{X} \dot{\eta}(t) + \mathbf{X}^T \mathbf{K} \mathbf{X} \eta(t) = \mathbf{X}^T \mathbf{F}(t) \quad (21)$$

Using Eq.(18) and Eq. (19)

$$\mathbf{X}^T \mathbf{M} \mathbf{X} \ddot{\eta}(t) = \mathbf{I} \ddot{\eta}(t) = \ddot{\eta} \quad (22)$$

$$\frac{d}{\omega_f} \mathbf{X}^T \mathbf{K} \mathbf{X} \dot{\eta}(t) = \frac{d}{\omega_f} \Omega^2 \dot{\eta}(t) = d \Omega \dot{\eta} \quad (23)$$

$$\mathbf{X}^T \mathbf{K} \mathbf{X} \eta(t) = \Omega^2 \eta \quad (24)$$

therefore

$$\ddot{\eta} + d \Omega \dot{\eta} + \Omega^2 \eta = \mathbf{X}^T \mathbf{F} \quad (25)$$

Equation (25) is the modal model of uncoupled second order differential equations. The motion of the structure can be found from the modal amplitudes, $\eta(t)$, using Eq. (20).

C. DISCRETE-TIME MODEL

The discrete-time state space model is found by solving the continuous-time equations. The *ith* equation of motion is

$$\ddot{\eta}_i(t) + d \omega_{oi} \dot{\eta}_i(t) + \omega_{oi}^2 \eta_i(t) = \mathbf{X}_i^T \mathbf{F}(t) \quad (26)$$

\mathbf{X}_i^T = transpose of the *ith* mode shape vector

$\mathbf{F}(t)$ = torquing force applied at a point

The homogeneous solution ($\mathbf{F}(t) = 0$) for Eq. (26) is [Ref. 4: p. 475-476]

$$\eta(t) = C_1 e^{-\gamma t} \cos(\omega_d t) + C_2 e^{-\gamma t} \sin(\omega_d t) \quad (27)$$

where

$$\gamma = \frac{d\omega_{oi}}{2} \quad (28)$$

$$\omega_d = \sqrt{\omega_{oi}^2 - \gamma^2} \quad (29)$$

The constants in Eq. (27) can be found by taking the derivative

$$\dot{\eta}(t) = (C_2 \omega_d - C_1 \gamma) e^{-\gamma t} \cos(\omega_d t) - (C_1 \omega_d + C_2 \gamma) e^{-\gamma t} \sin(\omega_d t) \quad (30)$$

and evaluating at $t = 0$

$$\eta(0) = C_1 \quad (31)$$

$$\dot{\eta}(0) = C_2 \omega_d - C_1 \gamma \quad (32)$$

Solving for C_1 and C_2

$$C_1 = \eta(0) \quad (33)$$

$$C_2 = \frac{\dot{\eta}(0)}{\omega_d} + \frac{\eta(0)\gamma}{\omega_d} \quad (34)$$

In matrix form

$$\begin{bmatrix} C_1 \\ C_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ \frac{\gamma}{\omega_d} & \frac{1}{\omega_d} \end{bmatrix} \begin{bmatrix} \eta(0) \\ \dot{\eta}(0) \end{bmatrix} \quad (35)$$

Rewriting Eq. (27) and Eq. (30) in matrix form

$$\begin{bmatrix} \eta(t) \\ \dot{\eta}(t) \end{bmatrix} = \begin{bmatrix} e^{-\gamma t} \cos(\omega_d t) & e^{-\gamma t} \sin(\omega_d t) \\ e^{-\gamma t} [\gamma \cos(\omega_d t) + \omega_d \sin(\omega_d t)] & e^{-\gamma t} [\omega_d \cos(\omega_d t) - \gamma \sin(\omega_d t)] \end{bmatrix} \begin{bmatrix} C_1 \\ C_2 \end{bmatrix} \quad (36)$$

Substituting Eq. (35) into Eq. (36), the solution can be written in terms of the initial conditions

$$\begin{bmatrix} \eta(t) \\ \dot{\eta}(t) \end{bmatrix} = \begin{bmatrix} e^{-\gamma t} \left[\cos(\omega_d t) + \frac{\gamma}{\omega_d} \sin(\omega_d t) \right] & \frac{1}{\omega_d} e^{-\gamma t} \sin(\omega_d t) \\ -\frac{\omega_o}{\omega_d} e^{-\gamma t} \sin(\omega_d t) & e^{-\gamma t} \left[\cos(\omega_d t) - \frac{\gamma}{\omega_d} \sin(\omega_d t) \right] \end{bmatrix} \begin{bmatrix} \eta(0) \\ \dot{\eta}(0) \end{bmatrix} \quad (37)$$

Letting

$$\mathbf{X}_i(t) = \begin{bmatrix} \eta_i(t) \\ \dot{\eta}_i(t) \end{bmatrix} \quad (38)$$

and

$$\mathbf{A}_i = \begin{bmatrix} e^{-\gamma t} \left[\cos(\omega_d t) + \frac{\gamma}{\omega_d} \sin(\omega_d t) \right] & \frac{1}{\omega_d} e^{-\gamma t} \sin(\omega_d t) \\ -\frac{\omega_o}{\omega_d} e^{-\gamma t} \sin(\omega_d t) & e^{-\gamma t} \left[\cos(\omega_d t) - \frac{\gamma}{\omega_d} \sin(\omega_d t) \right] \end{bmatrix} \quad (39)$$

the solution can be written as

$$\mathbf{X}_i(t) = \mathbf{A}_i(t) \mathbf{X}_i(0) \quad (40)$$

where \mathbf{A}_i is the state transition matrix of the i th mode. The non-homogeneous solution is

$$\mathbf{X}_i(t) = \mathbf{A}_i(t) \mathbf{X}_i(0) + \mathbf{B}_i x_i^T \mathbf{F}(0) \quad (41)$$

where the discrete-time input matrix, for constant \mathbf{F} , is given by

$$\mathbf{B}_i = \int_0^T \mathbf{B}_i(\tau) \Gamma \partial \tau \quad (42)$$

and $\Gamma = [0 \ 1]^T$ is the input matrix for the continuous-time system, and T is the sampling time. Solving Eq. (42) yields

$$\mathbf{B}_i = \begin{bmatrix} \frac{1}{\omega_o^2} [1 - e^{-\gamma T} \cos(\omega_d T) - \frac{\gamma}{\omega_d} e^{-\gamma T} \sin(\omega_d T)] \\ \frac{1}{\omega_d} e^{-\gamma T} \sin(\omega_d T) \end{bmatrix} \quad (43)$$

The discrete-time state equation for the i th equation of motion can be written as

$$\mathbf{X}_i(kT + 1) = \mathbf{A}_i(T) \mathbf{X}_i(kT) + \mathbf{B}_i(T) x_i^T \mathbf{F}(kT) \quad (44)$$

where \mathbf{A}_i and \mathbf{B}_i are evaluated at $t = T$. Here,

\mathbf{X}_i = vector of the i th modal amplitude and the i th modal velocity

- A_i = i th state transition matrix
- B_i = i th input vector
- x_i^T = transpose of the i th mode shape vector
- F = distributed force on the plant
- T = sampling time
- k = time index

Equation (44) can be expanded to include the disturbance input, $w(kT)$:

$$X_i(kT + 1) = A_i(T)X_i(kT) + B_i(T)x_i^T[F(kT) + w(kT)] \quad (45)$$

Equation (45) is the discrete-time mathematical model describing the motion of the structure in terms of its natural modes of vibration.

III. THE OBSERVER

A. INTRODUCTION

The observer design will be required to estimate the modal states in a noisy environment. Kalman filtering is the most widely used technique for accomplishing the production of state estimates in a noisy environment [Ref. 5: p.159]. The steady state Kalman filter was selected to minimize the computations during the actual plant observer operation. Use of a steady-state gain matrix for the observer allows the matrix to be computed separately from the operational observer, reducing the computer power required for the observer and allowing the algorithm to operate more rapidly.

B. KALMAN FILTER EQUATIONS

The discrete Kalman filter provides state estimates for the following dynamic system [Ref. 5: p. 159-162],

$$\mathbf{X}(k+1) = \mathbf{A}\mathbf{X}(k) + \mathbf{B}\mathbf{U}(k) + \mathbf{B}_n\mathbf{W}(k) \quad (46)$$

$$\mathbf{Y}(k+1) = \mathbf{C}\mathbf{X}(k+1) + \mathbf{V}(k+1) \quad (47)$$

\mathbf{X} = $n \times 1$ state vector

\mathbf{U} = $p \times 1$ control vector

\mathbf{W} = $r \times 1$ plant noise vector

\mathbf{Y} = $m \times 1$ measurement vector

\mathbf{V} = $m \times 1$ measurement noise vector

\mathbf{A} = $n \times n$ state transition matrix

\mathbf{B} = $n \times p$ control input matrix

\mathbf{B}_n = $n \times r$ plant noise input matrix

\mathbf{C} = $m \times n$ measurement matrix

The plant noise vector $\mathbf{W}(k)$ is gaussian white noise with

$$\mathbf{E}\{\mathbf{W}(k)\} = \mathbf{0} \quad (48)$$

$$\mathbf{E}\{\mathbf{W}(k)\mathbf{W}^T(k)\} = \mathbf{Q} \quad (49)$$

for all $k = 0, 1, 2, \dots$, and \mathbf{Q} is a positive semi-definite $r \times r$ matrix. $\mathbf{V}(k)$ is gaussian white noise with

$$\mathbf{E}\{\mathbf{V}(k)\} = \mathbf{0} \quad (50)$$

$$\mathbf{E}\{\mathbf{V}(k)\mathbf{V}^T(k)\} = \mathbf{R} \quad (51)$$

for all $k = 0, 1, 2, \dots$, and \mathbf{R} is a positive definite $m \times m$ matrix. The two random processes $\mathbf{W}(k)$ and $\mathbf{V}(k)$ are assumed to be independent, so that

$$\mathbf{E}\{\mathbf{V}(j)\mathbf{W}(k)\} = \mathbf{0} \quad (52)$$

for all $j = 1, 2, \dots$, and $k = 0, 1, 2, \dots$. The initial state $\mathbf{X}(0)$ is assumed to be a gaussian random vector with

$$\mathbf{E}\{\mathbf{X}(0)\} = \mathbf{0} \quad (53)$$

It is assumed that $\mathbf{X}(0)$ is independent of $\mathbf{W}(k)$ and $\mathbf{V}(k)$.

The optimal estimate of $\mathbf{X}(k+1)$ is denoted $\hat{\mathbf{X}}(k+1 | k+1)$. The Kalman filter is designed to minimize

$$\mathbf{J} = \mathbf{E}\{[\mathbf{X}(k+1) - \hat{\mathbf{X}}(k+1 | k+1)]^T [\mathbf{X}(k+1) - \hat{\mathbf{X}}(k+1 | k+1)]\} \quad (54)$$

The recursive realtions for generating $\hat{\mathbf{X}}(k+1 | k+1)$ are

$$\hat{\mathbf{X}}(k+1 | k) = \mathbf{A}\hat{\mathbf{X}}(k | k) + \mathbf{B}\mathbf{U}(k) \quad (55)$$

$$\hat{\mathbf{X}}(k+1 | k+1) = \hat{\mathbf{X}}(k+1 | k) + \mathbf{G}(k+1)[\mathbf{Y}(k+1) - \mathbf{C}\hat{\mathbf{X}}(k+1 | k)] \quad (56)$$

for $k = 0, 1, 2, \dots$, where $\hat{\mathbf{X}}(0 | 0) = \mathbf{0}$. $\hat{\mathbf{X}}(0 | 0)$ is set equal to zero since the expectation of $\mathbf{X}(0)$ is zero.

$\mathbf{G}(k+1)$ is an $n \times m$ matrix, called the Kalman Gain Matrix which is specified by the realtions:

$$\mathbf{P}(k+1 | k) = \mathbf{A}\mathbf{P}(k | k)\mathbf{A}^T + \mathbf{B}\mathbf{n}\mathbf{Q}(k)\mathbf{B}\mathbf{n}^T \quad (57)$$

$$\mathbf{G}(k+1) = \mathbf{P}(k+1 | k)\mathbf{C}^T[\mathbf{C}\mathbf{P}(k+1 | k)\mathbf{C}^T + \mathbf{R}(k+1)]^{-1} \quad (58)$$

$$\mathbf{P}(k+1 | k+1) = [\mathbf{I} - \mathbf{G}(k+1)\mathbf{C}]\mathbf{P}(k+1 | k) \quad (59)$$

$P(k | k)$ is the covariance matrix of the error between the states and their estimates

$$P(k | k) = E\{[X(k) - \hat{X}(k | k)][X(k) - \hat{X}(k | k)]^T\} \quad (60)$$

Since we are using the steady state gains the choice of $P(0 | 0)$ is irrelevant. $P(0 | 0)$ is initialized to zero in the gain derivation program for simplicity [Ref. 6: p. 139-140].

C. STEADY-STATE SOLUTION

If Equations (57), (58), and (59) are repeatedly iterated, $G(k + 1)$ will converge to a steady state value [Ref. 7: p. 263].

$$G_{ss} = \lim_{k \rightarrow \infty} G(k + 1) \quad (61)$$

The values of G_{ss} (or G) can be substituted into Eq. (56) making the steady state Kalman filter

$$\hat{X}(k + 1 | k) = A\hat{X}(k | k) + BU(k) \quad (62)$$

$$\hat{X}(k + 1 | k + 1) = \hat{X}(k + 1 | k) + G[Y(k + 1) - C\hat{X}(k + 1 | k)] \quad (63)$$

D. OBSERVER PERFORMANCE

The performance of an observer is judged by how accurately and rapidly it estimates the desired states. The performance measure of the observer as a whole is shown in equation (54). The normalized performance of the observer for individual states is

$$J_i = E[(x_i - \hat{x}_i)^2] / E[x_i^2] \quad (64)$$

which can be found using Eq. (65)

$$J_i = \sum_{k=0}^{\infty} (x_i(k) - \hat{x}_i(k))^2 T_s \div \sum_{k=0}^{\infty} x_i^2(k) T_s \quad (65)$$

J_i = performance measure for the i th state

$x_i(k)$ = value of the i th state at k

$\hat{x}_i(k)$ = observer estimate of i th state at k

T_i = sample interval

A normalized performance measure is used to aid comparison of the performance of the observer in estimating various states. From Eq. (65) it can be shown that if $\hat{x}_i(k) = 0$ for all $k = 0, 1, 2, \dots$ that J_i would be unity. Therefore, the better the performance of the observer, the smaller the fraction of one J_i will be.

IV. SIMULATION

A. INTRODUCTION

The objectives of the simulation were to

- determine the sensitivity of the observer performance and settling time to changes in the ratio of plant noise to measurement noise,
- determine the effect on observer performance and settling time of increasing the number of modes observed in the matched plant/observer, and
- determine the performance for the reduced order observer.

B. PLANT AND OBSERVER DATA

The dynamic model is a truncated form of a preliminary space station configuration; the phase II dual keel structure.¹ The full model consists of an infinite number of natural modes but this was restricted to the first ten active modes for this study due to limitations on computer resources. As will be shown reasonable data can be obtained with this simplification in examining the observer performance.

C. SIMULATION PROGRAMS

The simulation was broken down into two segments due to the large memory and computational time requirements. The first program computed the steady state observer gain matrix (G). The second program ran the observer and the plant when the plant was subjected to an impulse excitation.

The steady state observer gain matrix (G) was obtained by repeated iteration of equations (57), (58), and (59). The equations were run until the values of the matrix changed by less than a set fraction. The following formula was used to check the changes in the gain matrix elements

$$\Delta g_{ij} = [g_{ij}(k+1) - g_{ij}(k)]/g_{ij}(k+1) \quad (66)$$

The program was terminated when δg_{ij} was less than 10^{-10} .

The settling time for the estimates of the states to be within 2% of the actual states was determined by finding the eigenvalues of $A - G^*C$ then computing as follows [Ref. 6: p. 139-143]

¹ The model for preliminary station configuration was provided courtesy of McDonnell Douglas Astronautics Company, 5301 Bolsa Avenue, Huntington Beach, CA 92647.

$$T_s = \log(.02) / \log(\lambda_{AGC_{min}}) \quad (67)$$

The expected error in the sensor, i.e., the standard deviation of the noise in the measurement, was chosen as 10^{-3} feet per sec. per sec. based on the natural frequencies in the structure and reasonable sensor sensitivity [Ref. 2: p. 79-80]. The expected plant noise was varied to find the range of ratios between plant and measurement noise that the filter would be effective. This approach was taken since the plant noise contributors are not currently well defined.

The second program subjected the plant as modeled in Eq. (45) to an impulse excitation and then had the observer estimate the selected states using observer equations (62) and (63). Observer performance was computed using Eq. (65).

A third program was used to find the contribution of unobserved modes to the noise in the Kalman observer. The program ran the plant subject to an impulse excitation and computed the product of the measurement matrix C times the unobserved modes of the state vector $X(k)$ for a measure of the noise contributed by the unobserved modes.

The three programs are listed in the appendices.

D. EFFECT OF PLANT TO MEASUREMENT NOISE RATIO ON OBSERVER PERFORMANCE

The ratio of the variance of the plant noise (PN) to the variance of the measurement noise (MN) was found to have a strong effect on the Kalman Observer performance (J) and settling time (T_s). Figures 1 through 6 show the observer performance for a 3 mode matched plant and observer system for progressively smaller PN/MN ratios.² Figure 7 shows the performance for the seventh mode (position) versus several values of PN/MN. Figure 8 is the settling time versus the same PN/MN ratios.

The figures show that, for all of the plotted performance values, the observer performance is at least marginally acceptable regardless of the PN/MN ratio. Decreasing the PN/MN ratio leads to an even more rapid degradation in observer performance. The settling times also rapidly increase as the PN/MN ratio decreases.

² Figures 1-6 and 9-15 show the performance measure for each mode. The bar for the mode position is immediately to the right of the numbered tick mark on the x-axis scale, the mode velocity is next to it immediately adjacent to the tick mark without a number.

E. EFFECTS OF INCREASED MODES ON OBSERVER PERFORMANCE

The matched plant/observer was run with increasing numbers of modes to see if there was an effect on observer performance (J) or settling time T_s . Figures 7 through 17 are of observer performance for systems with increasing numbers of modes in the system being observed. Figure 18 is of settling time versus the number of modes in the system. The ratio of PN/MN was kept constant at $PN/MN = 2.5 \times 10^9$.

The increasing of the number of modes for the matched plant/observer had negligible effect on the performance for the individual modes. The performance value for the modes was effectively constant. Settling times for the observers increased as the number of modes was increased.

F. REDUCED ORDER KALMAN OBSERVER

The Kalman Observer has been shown to be effective where the number of modes observed matches the number of modes in the plant. The Kalman Observer was then run with the one less mode observed than the number of modes in the plant. The gain matrix (G) from the matched system was used. The observer failed with the state estimates produced by the observer becoming excessively large and having settling times of hours vice minutes. Since the purpose of the observer was to provide estimates for use in controlling the plant the time delay makes the estimates unusable.

The cause of the observer failure is apparent when you look at the last portion of Eq. (56) of the Kalman Observer

$$G[Y(k+1) - \hat{C}\hat{X}(k+1|k)] \quad (68)$$

This portion of the observer equation is the correction of $\hat{X}(k+1|k)$ to produce $\hat{X}(k+1|k+1)$. The design of the Kalman observer is to produce an estimate despite the measurement noise but, with the reduced order filter there is additional unanticipated noise which causes over correction of the values of \hat{X} leading to the state estimates being excessively large and settling times being too long. This can be shown by examining what composes $Y(k+1) - \hat{C}\hat{X}(k+1|k)$

$$\begin{array}{c}
\begin{bmatrix}
x_1(k) \\
x_2(k) \\
\uparrow \\
\downarrow \\
x_{m-1}(k) \\
x_m(k) \\
\text{---} \\
x_{m+1}(k) \\
x_{m+2}(k) \\
\uparrow \\
\downarrow \\
x_{n-1}(k) \\
x_n(k)
\end{bmatrix}
- C
\begin{bmatrix}
\hat{x}_1(k) \\
\hat{x}_2(k) \\
\uparrow \\
\downarrow \\
\hat{x}_{m-1}(k) \\
x_m(k) \\
\text{---} \\
0 \\
\uparrow \\
\downarrow \\
0
\end{bmatrix}
\end{array}
\quad (69)$$

C times the state $x_{m-1}(k)$ through $x_n(k)$ is unanticipated noise so if

$$C
\begin{bmatrix}
x_1(k) \\
x_2(k) \\
\uparrow \\
\downarrow \\
x_{m-1}(k) \\
x_m(k)
\end{bmatrix}
- C
\begin{bmatrix}
\hat{x}_1(k) \\
\hat{x}_2(k) \\
\uparrow \\
\downarrow \\
\hat{x}_{m-1}(k) \\
\hat{x}_m(k)
\end{bmatrix}
= 0
\quad (70)$$

the remaining portion of the C matrix times the modal states is an equivalent noise.

Table (1) shows the growth of the unanticipated noise in the filter as the number of unobserved modes in the plant grows. Table (2) shows the individual contributions of the individual modes when left unobserved. Table (1) shows that the unanticipated noise is much larger than that expected by the filter (10^{-3}). Table (2) shows that there are modes that do not markedly contribute to the noise and that they might successfully be left unobserved if the measurement noise estimate was already much larger than these noise sources.

Table 1. CUMULATIVE UNANTICIPATED NOISE FROM UNOBSERVED MODES

Num- ber of Unob- served Modes	Unob- served Modes	E1	E2	E2
1	10	0.647	97.440	3.277
2	10-11	366.354	95.764	3.142
3	10-12	366.355	95.764	3.142
4	10-13	365.565	95.855	3.143
5	10-14	365.426	96.032	5.704
6	10-15	144195.7	116.205	2201.94
7	10-16	148006.8	170.142	5475.21
8	10-17	473974.3	39424.1	5692.50
9	10-18	474344.2	60078.8	9419.77
10	10-19	474358.5	68987.7	9865.27

Table 2. UNANTICIPATED NOISE FROM UNOBSERVED MODES BY MODE

Unobserved Mode	E1	E2	E3
10	0.64716	97.4403	3.27761
11	359.342	0.20929	0.21429
12	0.9353E-08	0.1364E-07	0.2196E-07
13	0.3852E-02	0.4409E-02	0.9017E-03
14	0.5615E-03	0.2592E-01	2.57554
15	143090.9	17.9682	2167.02
16	195.736	83.3458	5675.91
17	324471.2	39252.26	221.170
18	216.240	21133.6	3736.93
19	7.69298	8829.95	458.504
20	2.3194	3949.16	108.981

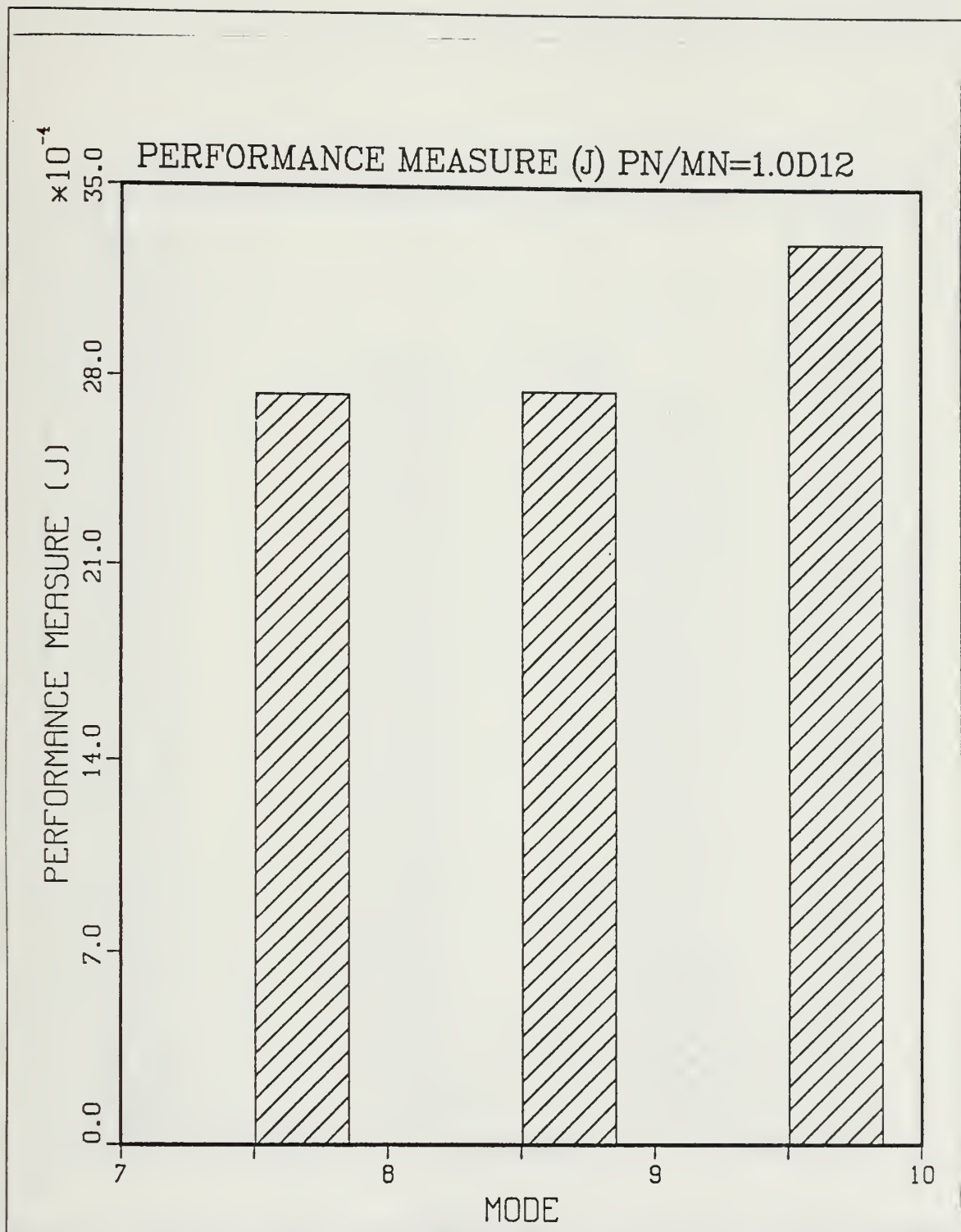


Figure 1. Observer Performance (J) PN/MN = 1.0d12

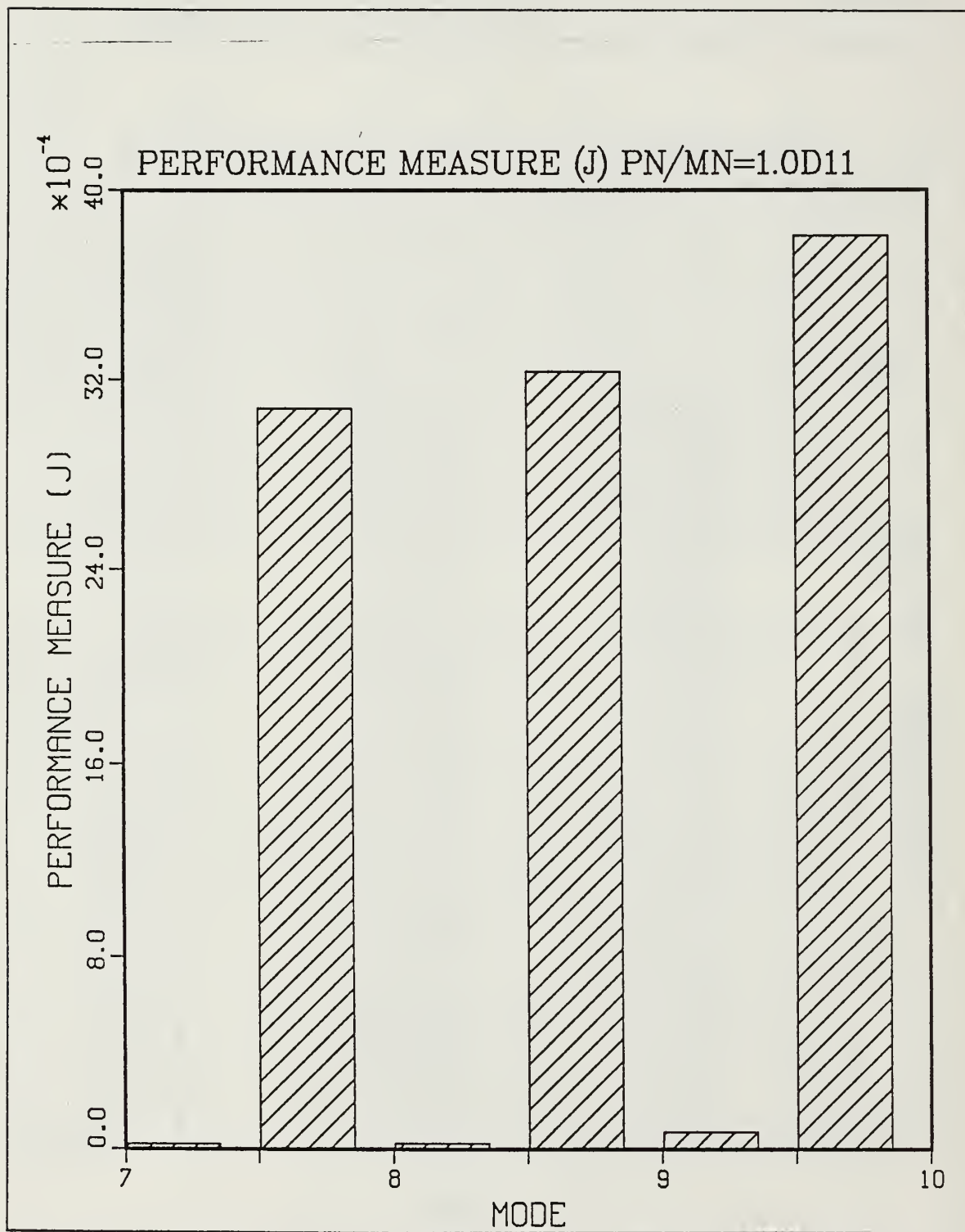


Figure 2. Observer Performance (J) PN/MN = 1.0d11

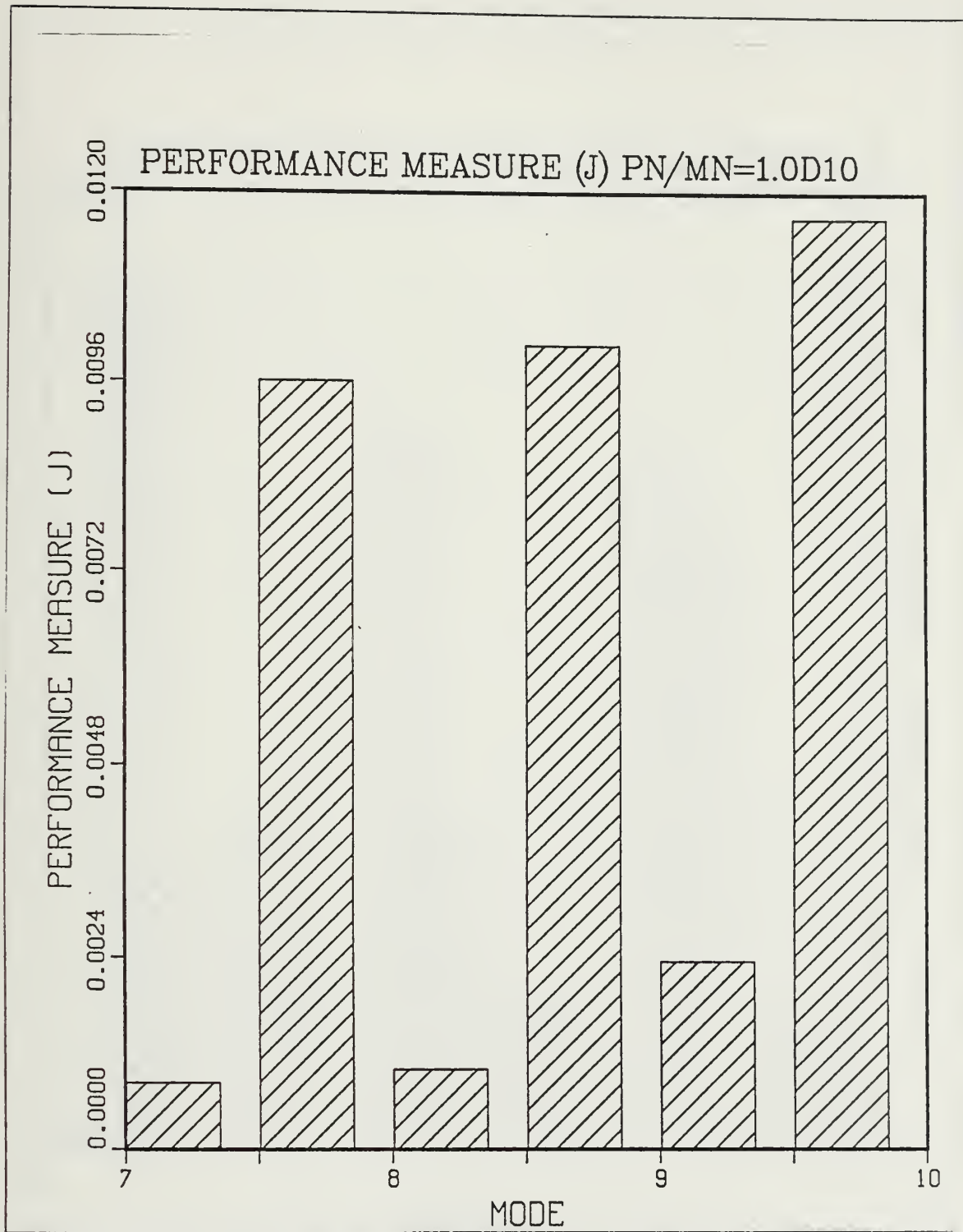


Figure 3. Observer Performance (J) PN/MN = 1.0d10

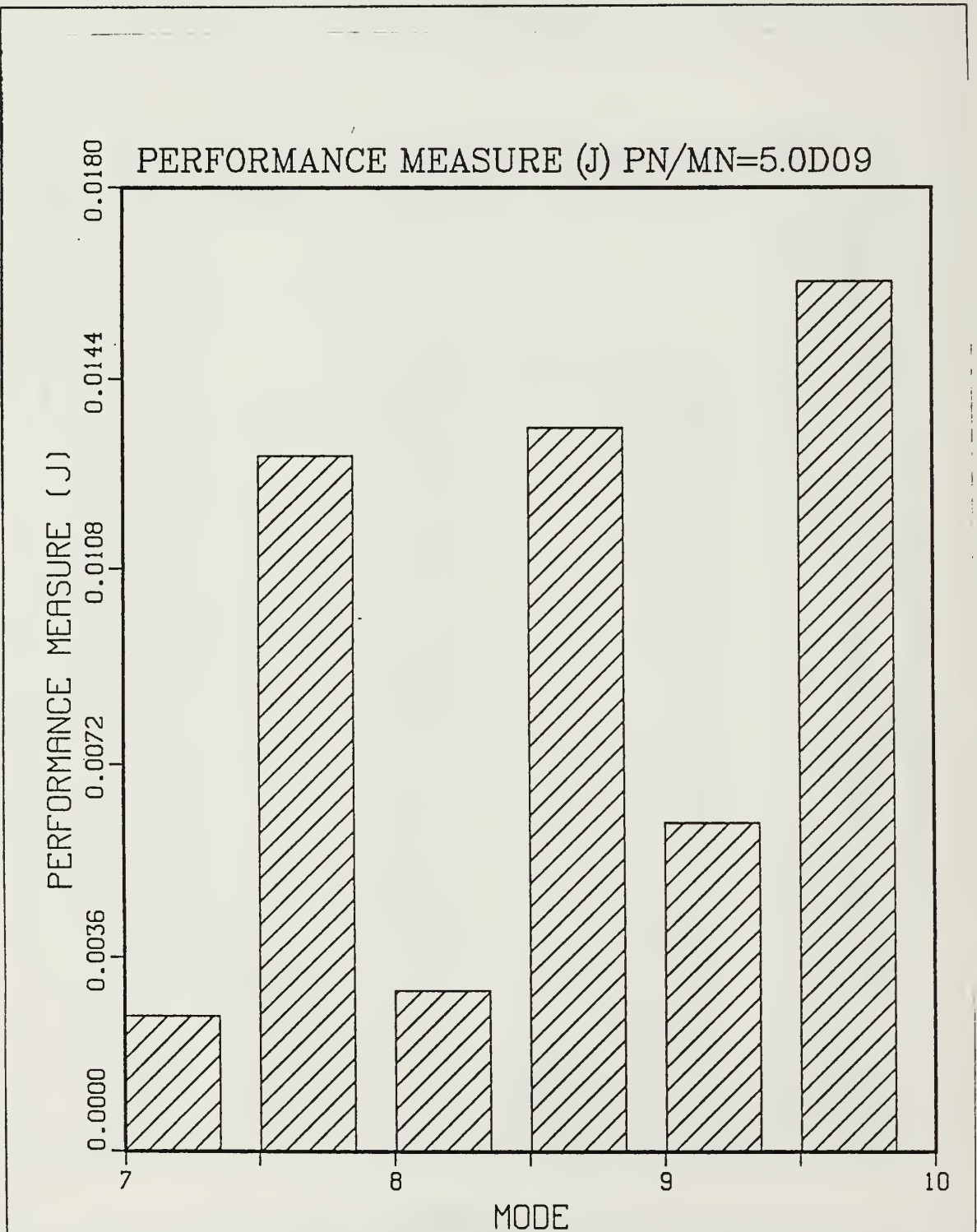


Figure 4. Observer Performance (J) PN/MN = 5.0d09

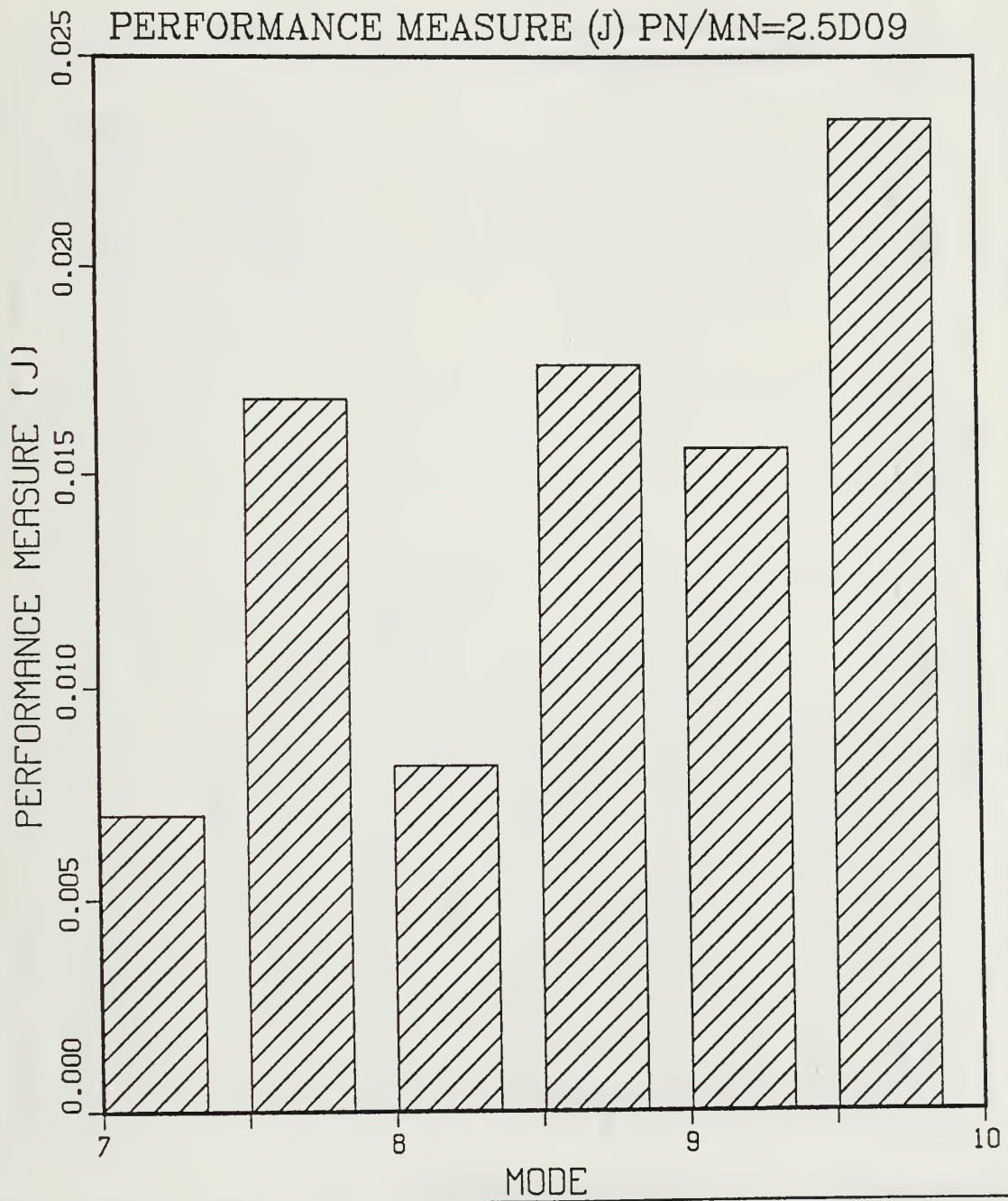


Figure 5. Observer Performance (J) PN/MN = 2.5d09

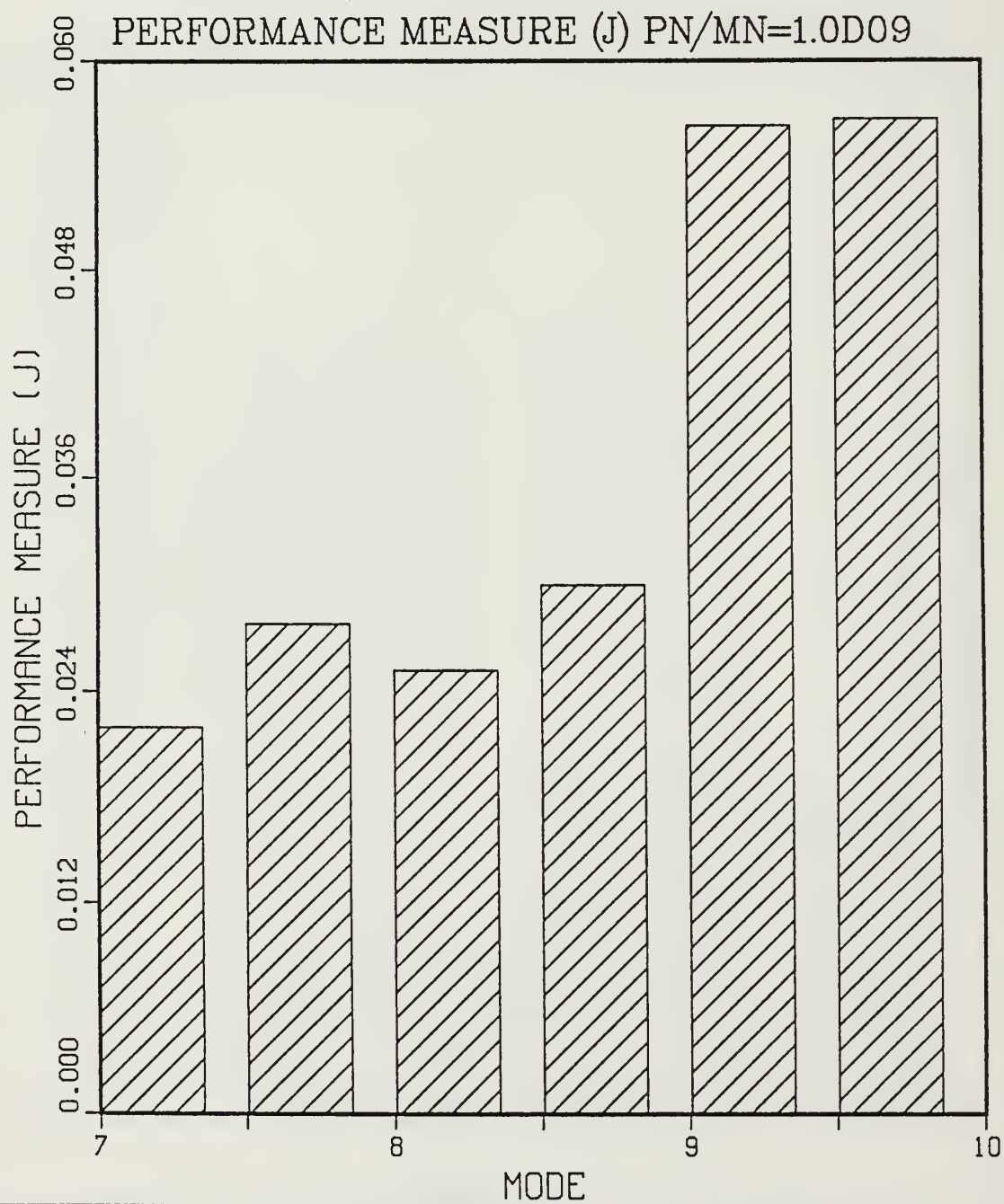


Figure 6. Observer Performance (J) PN/MN = 1.0d09

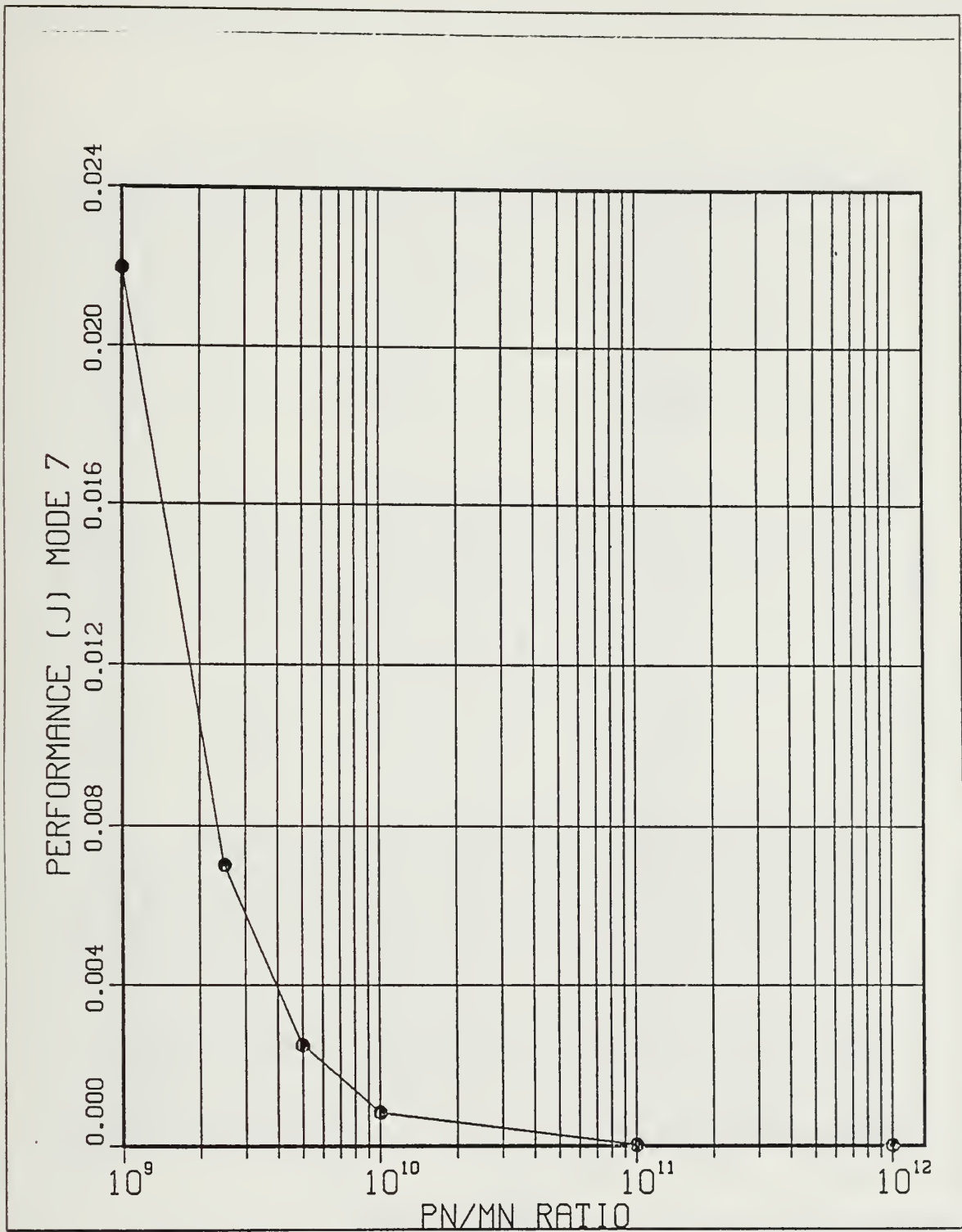


Figure 7. Mode 7 (Position) Observer Performance versus PN/MN

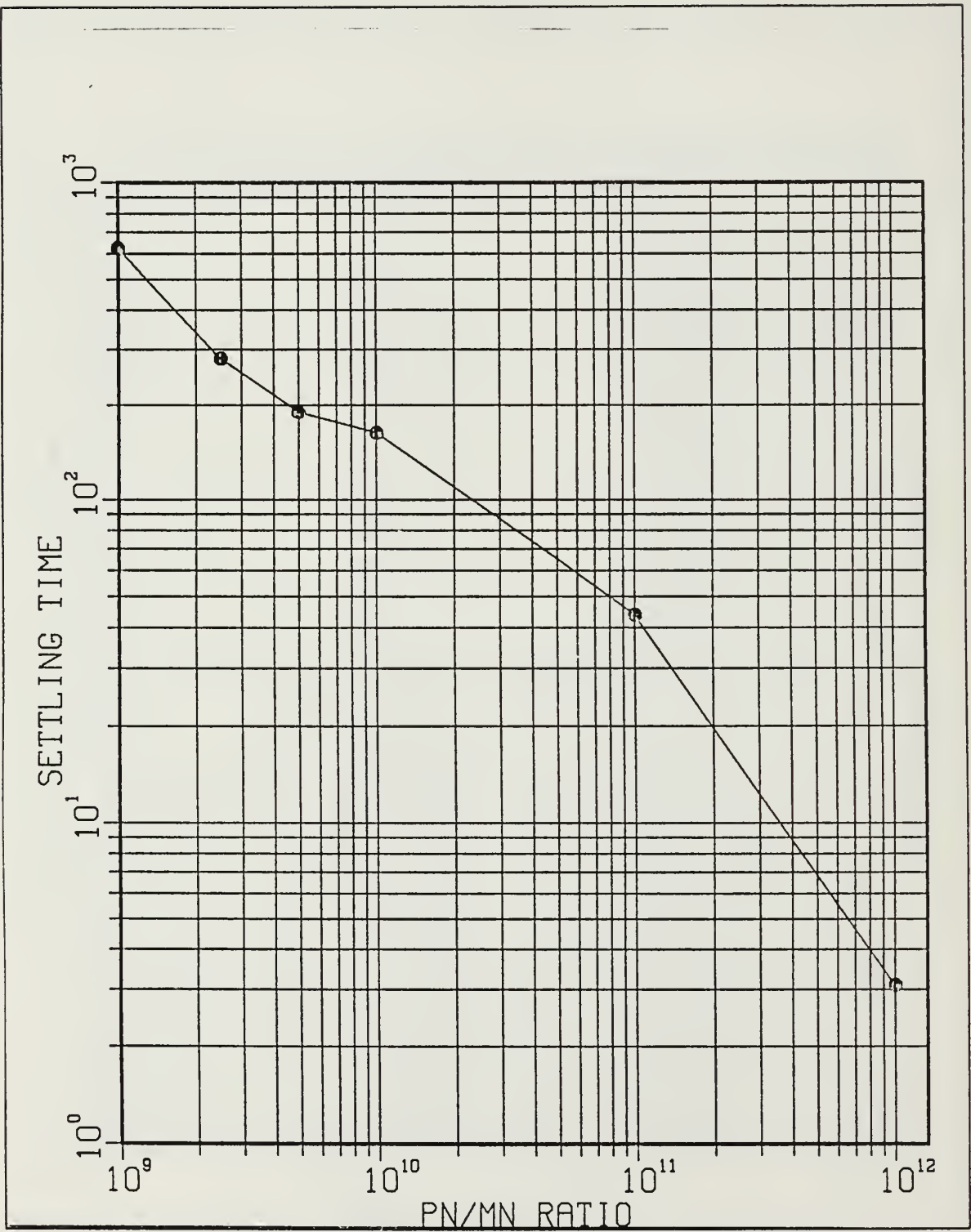


Figure 8. Settling Time versus PN/MN

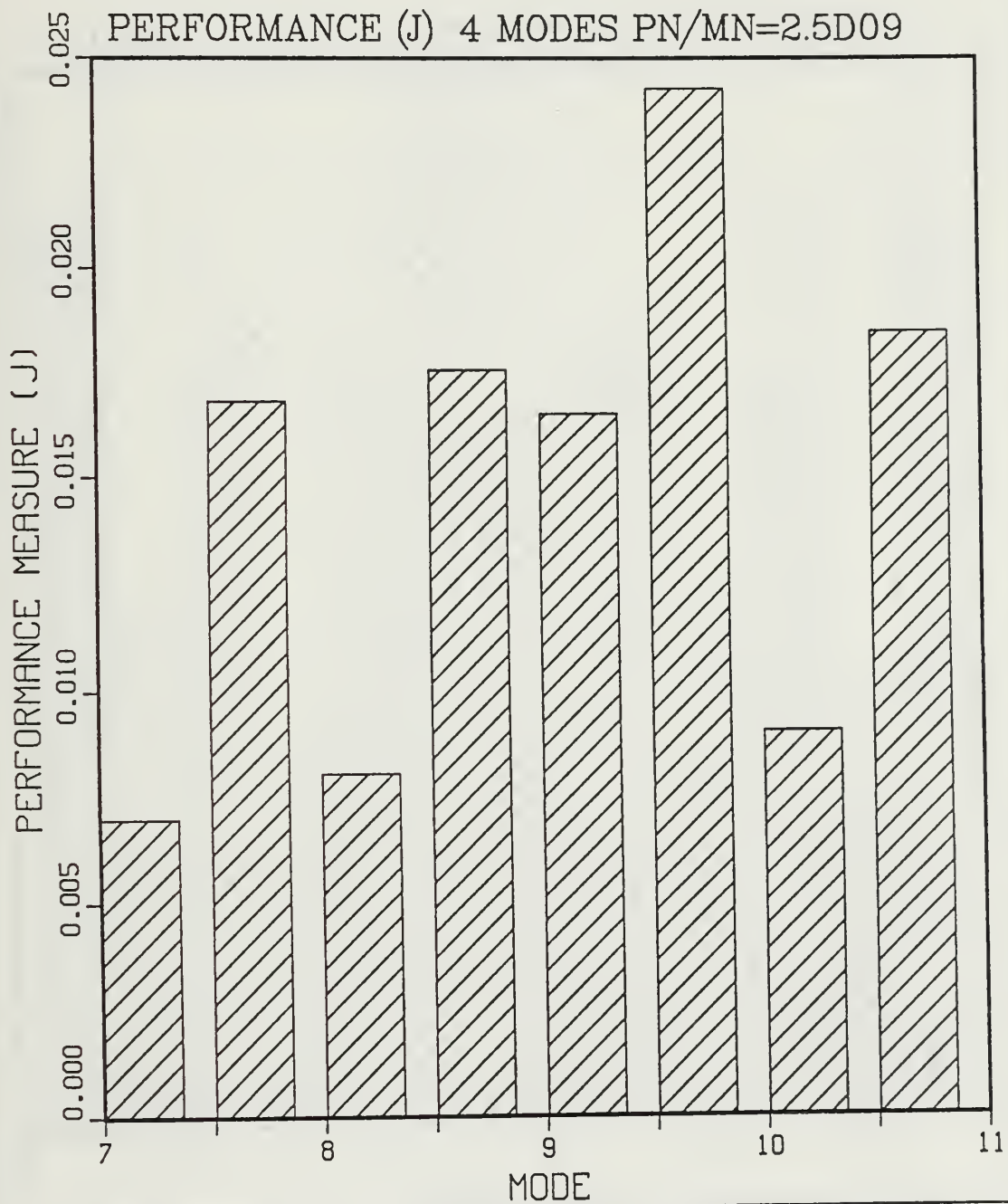


Figure 9. Observer Performance (J) 4 Modes (7 - 10)

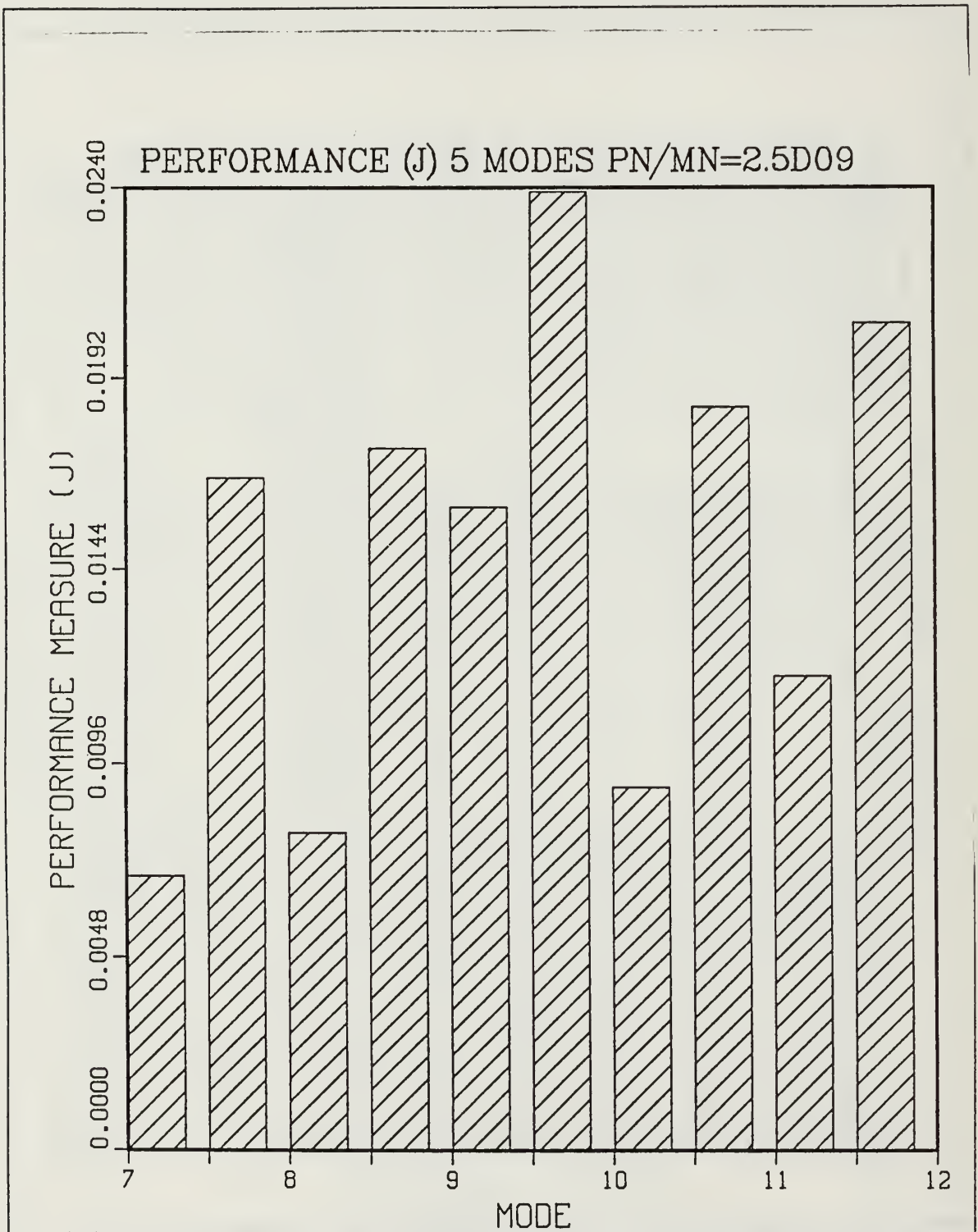


Figure 10. Observer Performance (J) 5 Modes (7 - 11)

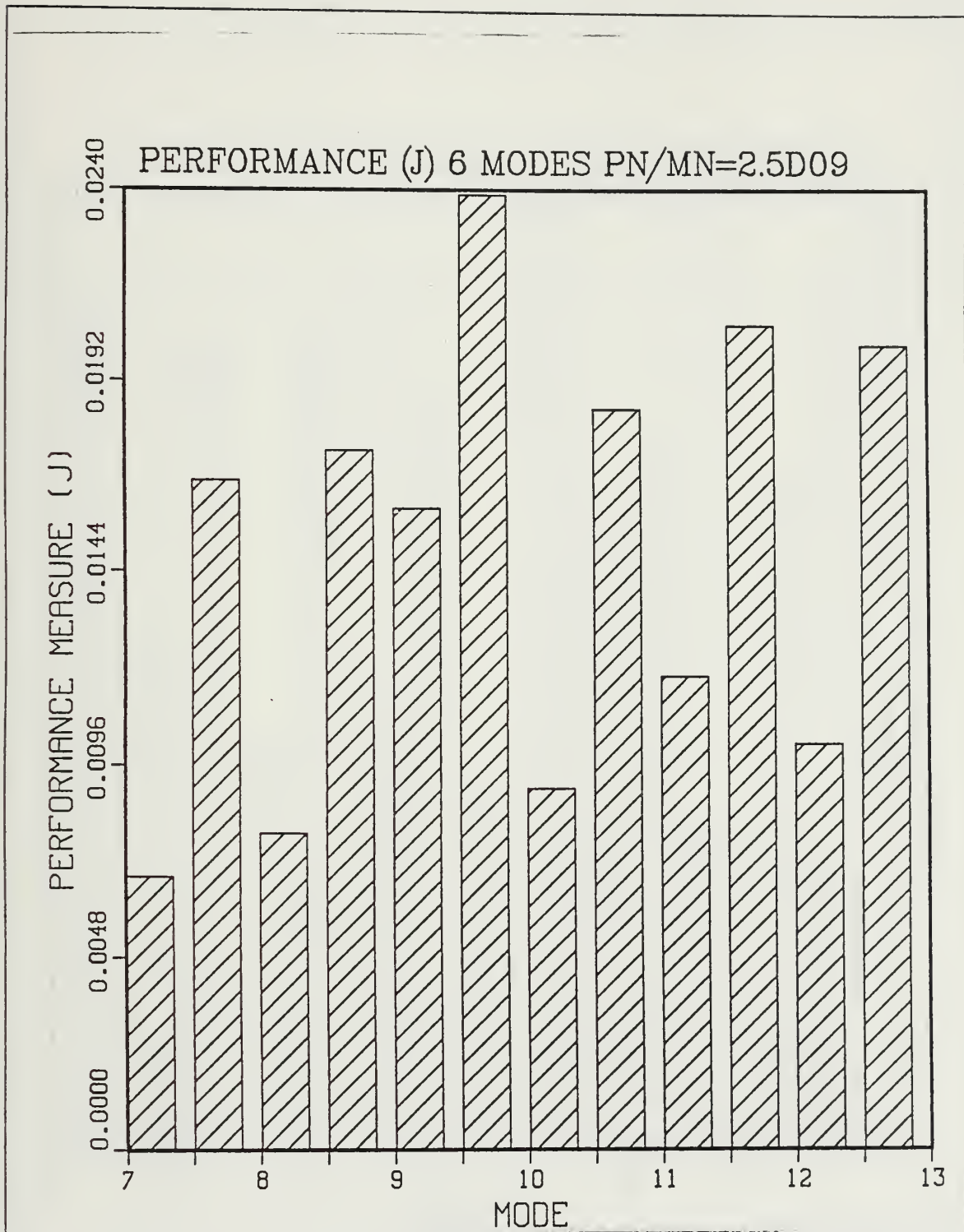


Figure 11. Observer Performance (J) 6 Modes (7 - 12)

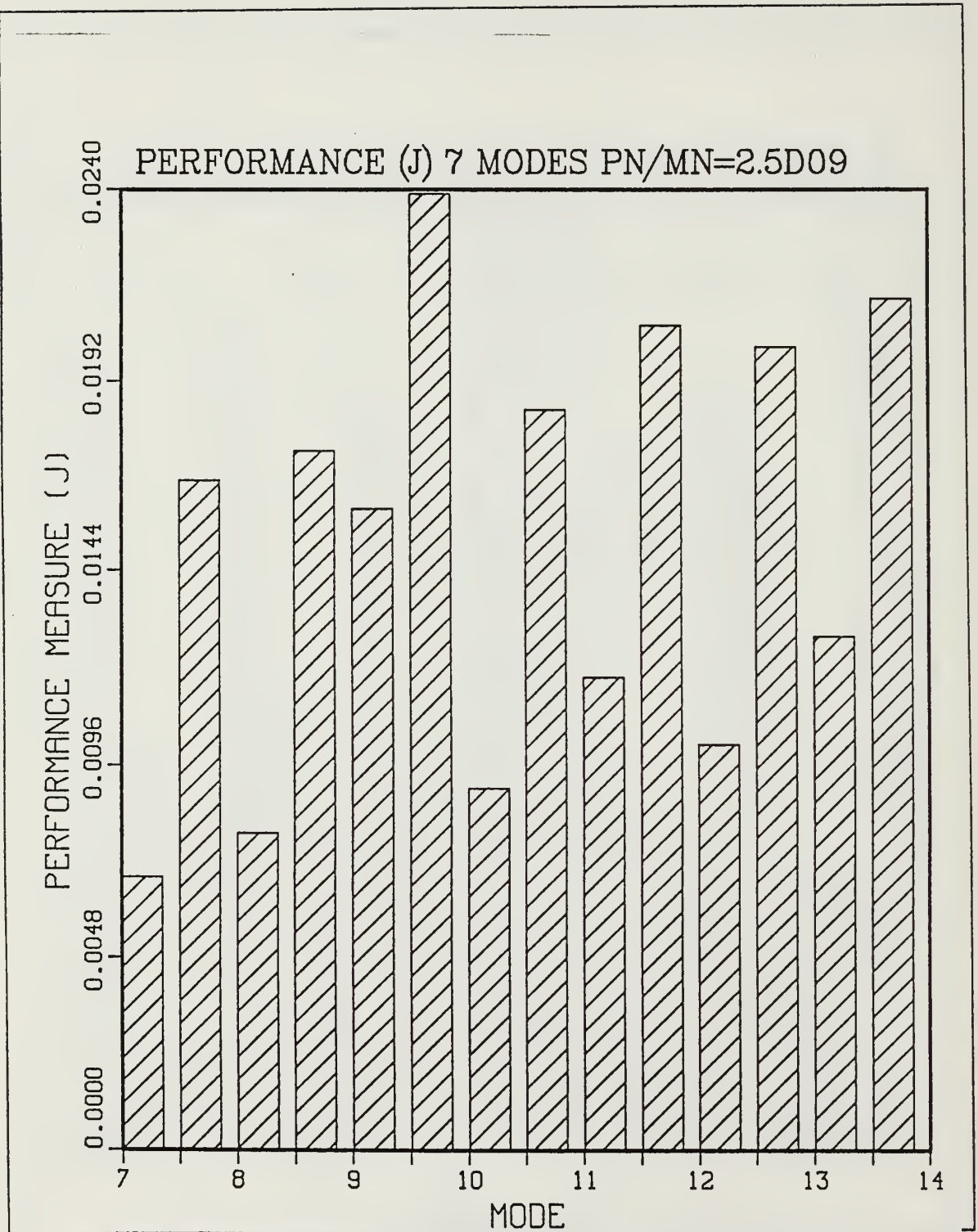


Figure 12. Observer Performance (J) 7 Modes (7 - 13)

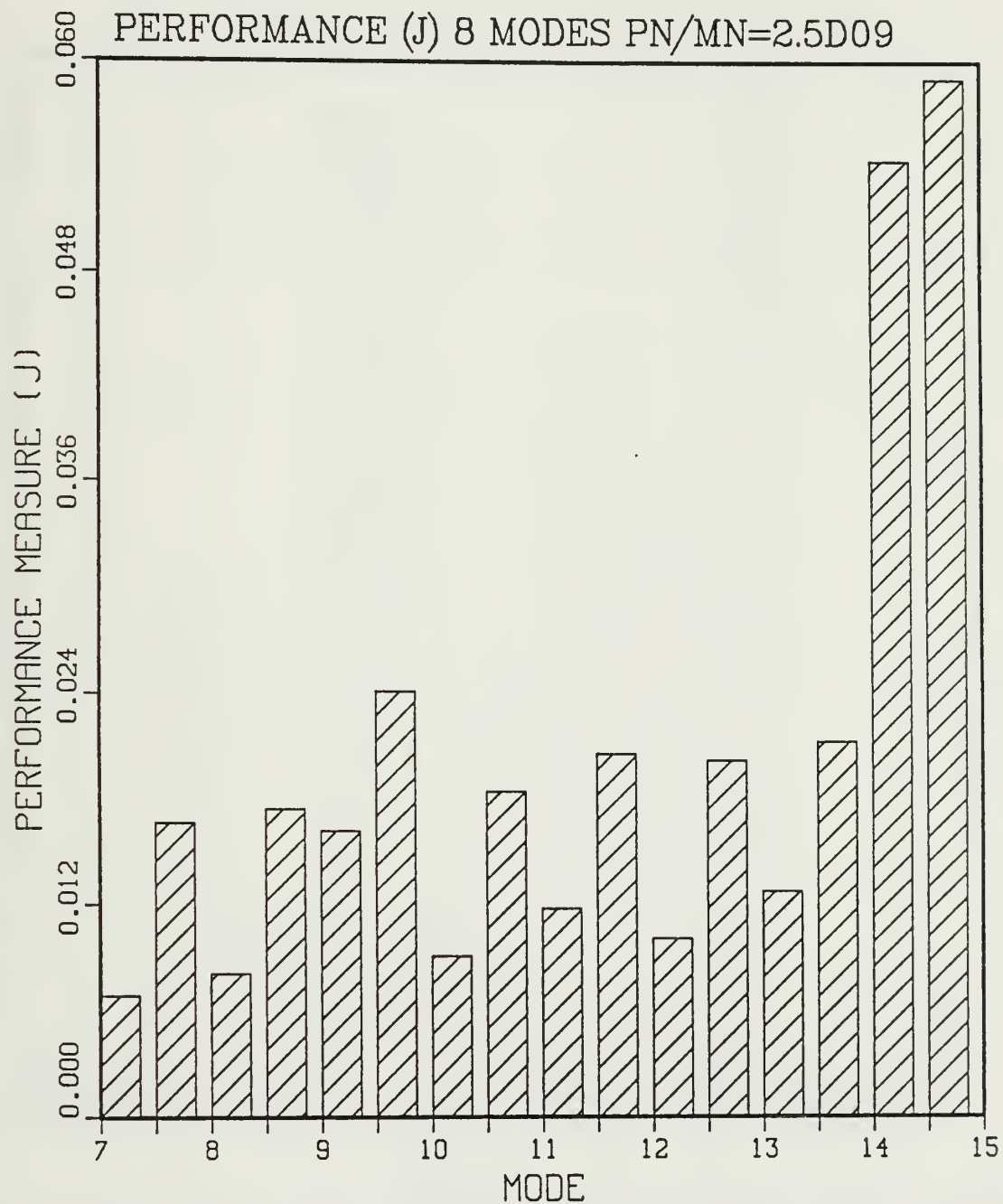


Figure 13. Observer Performance (J) 8 Modes (7 - 14)

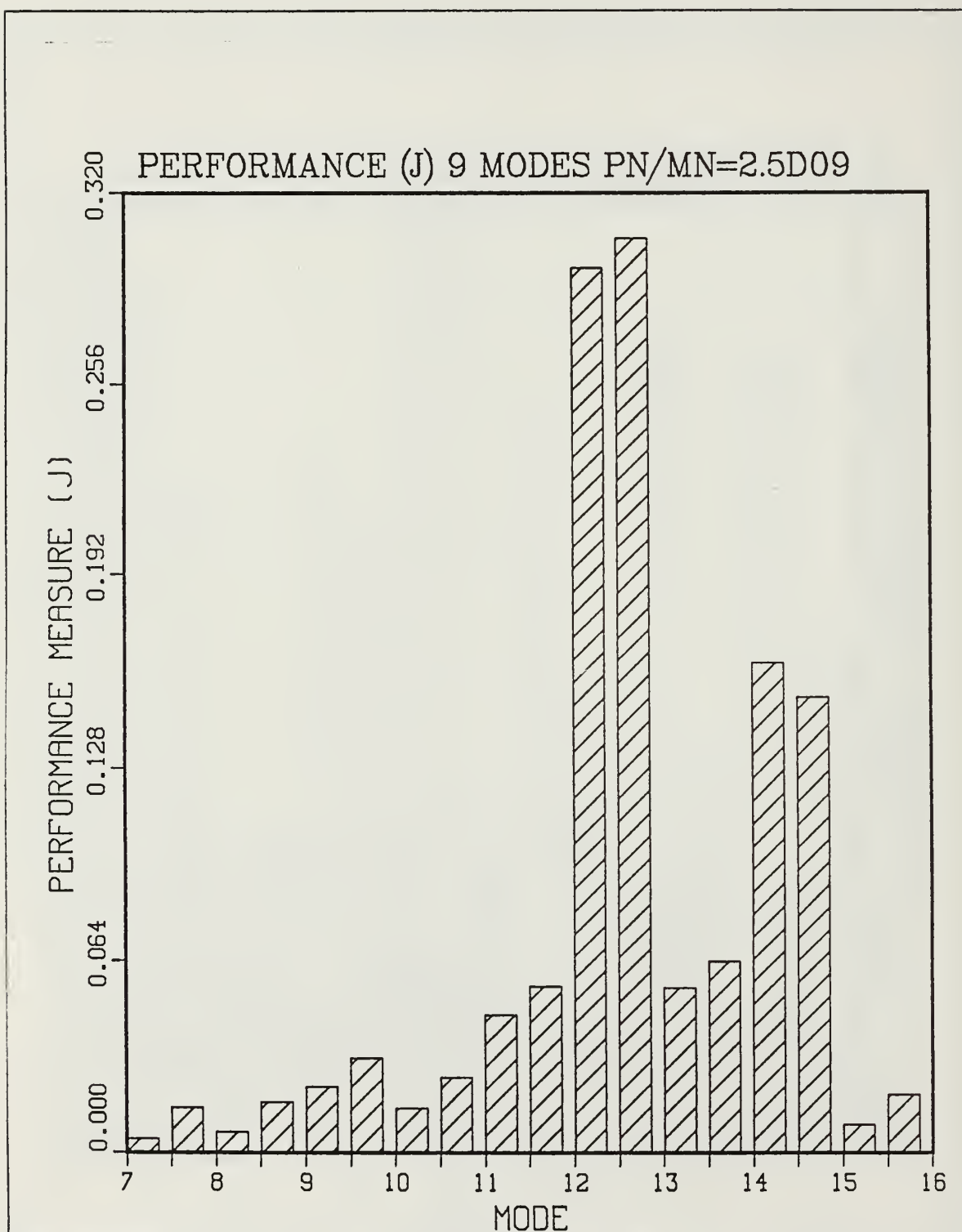


Figure 14. Observer Performance (J) 9 Modes (7 - 15)

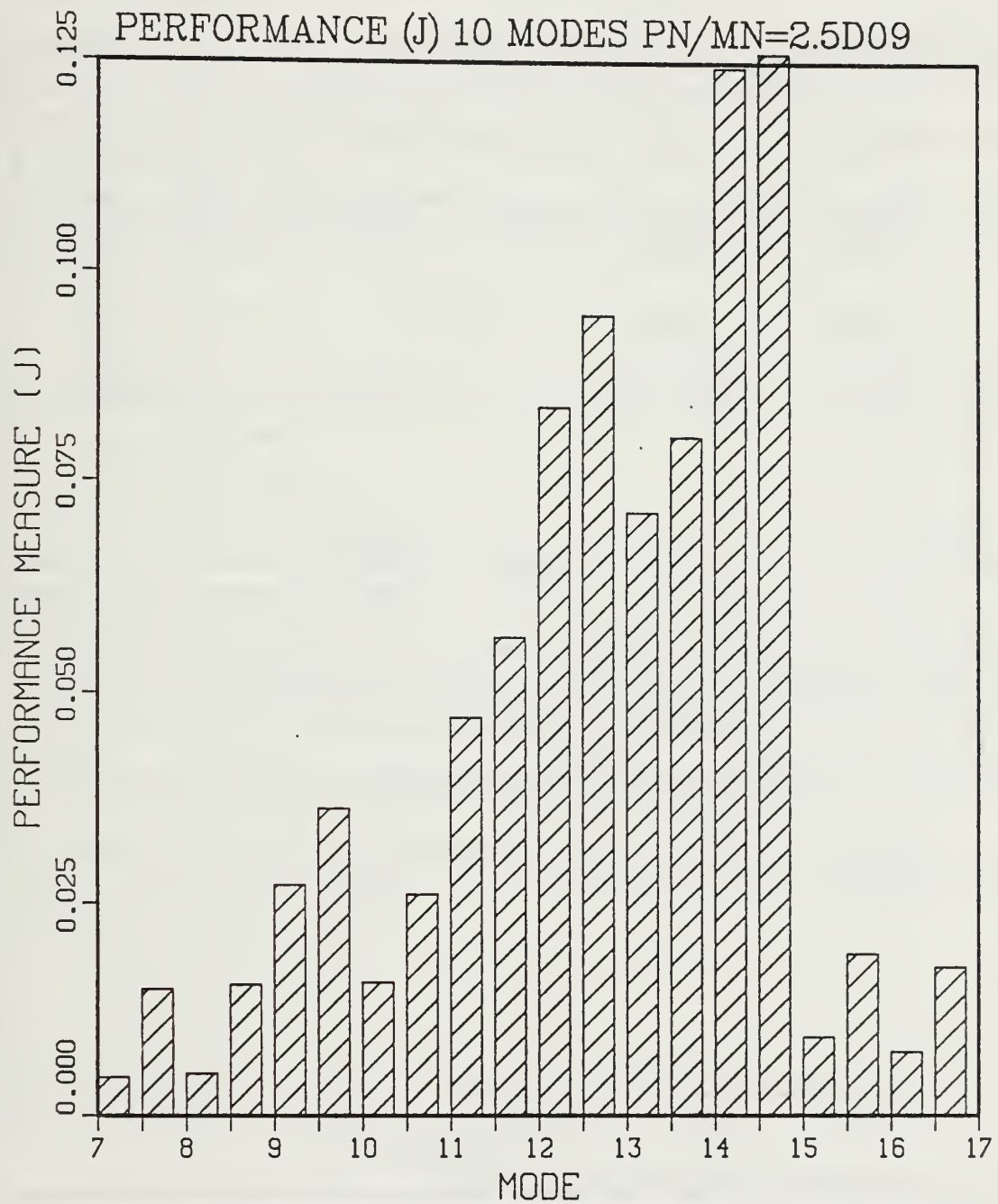


Figure 15. Observer Performance (J) 10 Modes (7 - 16)

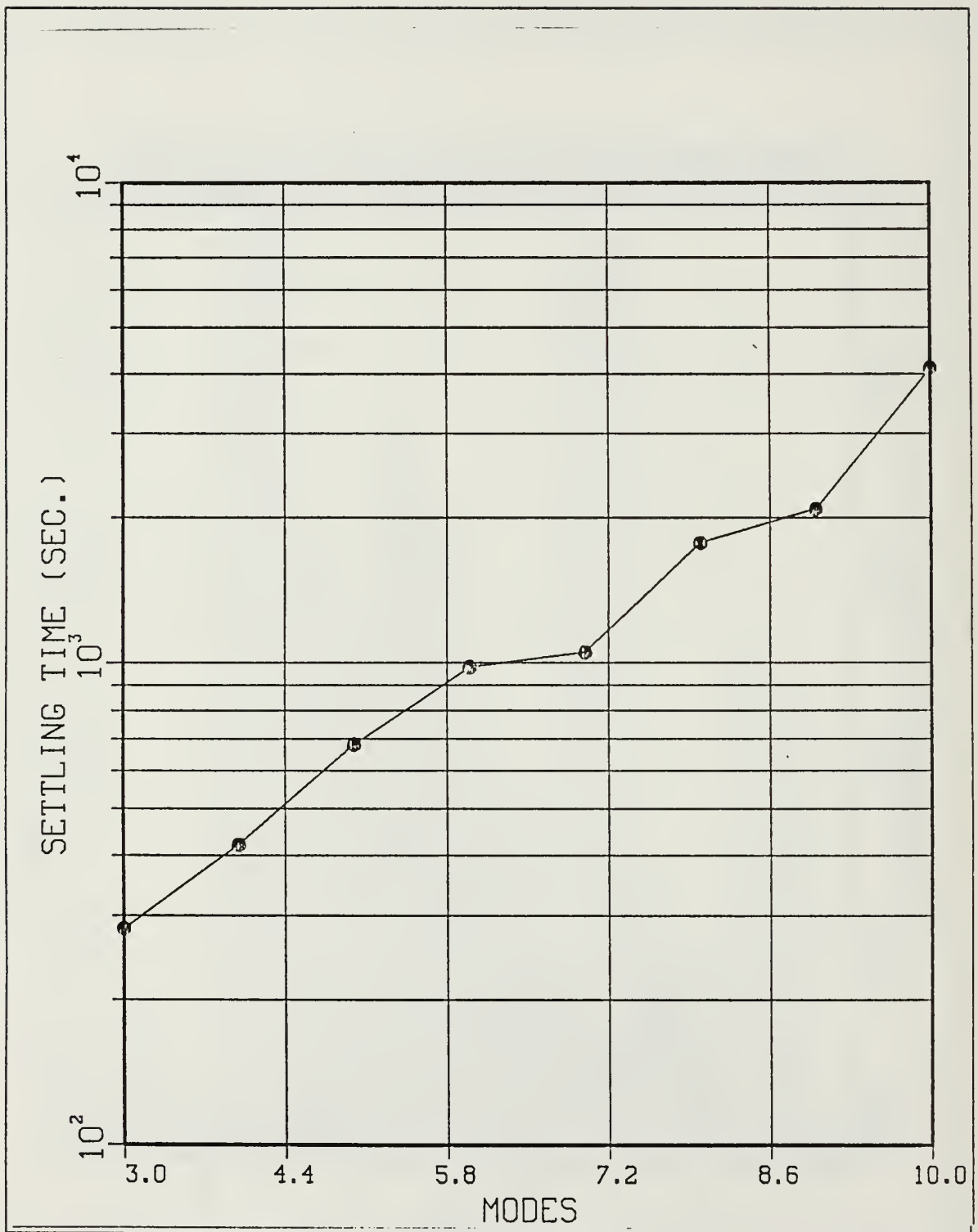


Figure 16. Settling Time versus number of Modes Observed

V. CONCLUSIONS AND RECOMMENDATIONS

A. CONCLUSIONS

Simulations runs showed that a matched plant/observer can work if the following criterions are meet:

- The ratio of plant noise to measurement noise is sufficiently high to produce a usable settling time.
- Sufficient computational power is available to run the matched observer. The amount of memory and number of computation goes up as the number of modes observed increases.

Utilizing a reduce order observer for an arbitrarily selected set of modes is not feasible. The non-observed modes add so much noise to the system that settling times and observer performance are so poor as to render the observer useless for obtaining state values for plant control.

B. RECOMENDATIONS

The work on the Kalman Observer for Large Space Structures lead to the following recommendations for further research:

- Identify those modes that contribute the largest noise to the Kalman observer and set the observer to estimate these states in addition to those required for plant control. A possible method for identifying the modes that contibute the largest noise to the observer would be the Karhunen-Loeve expansion.
- Modifying the plant/observer to model the use of sensors at additional positions to see if the increase in the data rate will help decrease settling time.
- Modify the model to incorporate noise injection into more than one location . The current model has noise injected at only one position, a useful simplification for initial analysis but not realistic.

APPENDIX A. KALMAN GAIN MATRIX GENERATION PROGRAM

```

C
C *****
C *****          GGAIN          *****
C ***** ADAPTED TO RUN KALMAN FILTER AND COMPUTE THE *****
C ***** G MATRIX BY ITERATION STOPPING WHEN THE *****
C ***** THE MATRIX GOES TO STEADY STATE *****
C *****
C *****
C *****
C *****
C *****
C *****          VARIABLE DECLARATIONS          *****
C *****
C *****
C
EXTERNAL STMTRX,DLINRG,EXCMS, DEVCGR
CHARACTER*6 NAM
CHARACTER*1 AGAIN,CORECT,RAGAIN
INTEGER ROWN1,ROWN2,ROWN3,COUNT,NODE,MODE,KQ,EMODE,SMODE,R2M,C2M
INTEGER CT,CF,KADJ,CFADJ,LOOP,PRNT,JJ,JK,N1,JR,KR,MR,ISEED,M2
INTEGER JL,J1,JM
REAL LAMA(100), UGVEX(684,100),RNODE,RMODE,MIN
REAL*8 PHI(2,2,100),GAMMA(2,100),EGT,GMA,WN,W1,X1T,X2T,TIME
REAL*8 A(200,200),B(200,3),F(3, 50),IMPLSE,ENERGY
REAL*8 COSW1T,SINW1T,X1(100),X2(100),COST(100)
REAL*8 DAMP,SAMPT,PI,SAMPTM,SUM1,SUM2,SUM3,SUMC
REAL*8 C(6,200), IDENT( 50, 50), RMN(6,6), QPN(3,3)
REAL*8 PK( 50, 50), Y(6), BN(200,3)
REAL*8 PNVARX, PNVARY, PNVARZ
REAL*8 MNVX1, MNVY1, MNVZ1, SUM, BQBT(50,50)
REAL*8 TMP1( 50,3), TMP2(3,3), TMP3( 50, 50)
REAL*8 PK1( 50, 50),G( 50,3)
REAL*8 DY(3), ES,ED,ESUM,CGN,PRT
REAL*8 SF, N9, TCHK, ACHK, H1, H2, H3, H4, H5, H6
REAL*8 AGC(100,100)
C
COMPLEX*8 EVAL(100), EVEC (100,100)
C
C
C
C *****
C *****          VARIABLE DEFINITIONS          *****
C *****
C *****
C
STMTRX = SUBROUTINE EXTABLISHES STATE TRANSITION MATRICIES
LAMA = VECTOR OF THE SQUARE OF THE NATURAL FREQUENCIES
UGVEX = MODE POSITONS AND SLOPES OF THE NODAL POINTS
PHI = STATE TRANSITION MATRICIES FOR EACH MODE
GAMMA = INPUT TRANSITION MATRIX
A = DIAGONAL MATRIX CONSISTING OF PHI
B = INPUT MATRIX OF GAMMA AND CONTROL SLOPES
DAMP = DAMPING FACTOR
SAMPT = SAMPLING TIME

```

C	TCX, TCY, TCZ = CONTROL TORQUE VALUES	GMA00520
C	ENERGY = TOTAL SYSTEM ENERGY	GMA00530
C	IMPLSE = IMPULSE INPUT FUNCTION	GMA00540
C	MIN = NUMBER OF MINUTES SYSTEM WILL BE OBSERVED	GMA00550
C	SMODE = NUMBER OF STARTING MODE (INT)	GMA00560
C	MODE = NUMBER OF MODES (INT)	GMA00570
C	EMODE = NUMBER OF THE LAST MODE (INT)	GMA00580
C	NODE = NUMBER OF THE NOISE INPUT MODE (INT)	GMA00590
C	*** NOISE SLOPE LOCATIONS IN DATA MATRIX ***	GMA00600
C	ROWN1 = X-SLOPE LOCATION	GMA00610
C	ROWN2 = Y-SLOPE LOCATION	GMA00620
C	ROWN3 = Z-SLOPE LOCATION	GMA00630
C	C = OUTPUT MATRIX FOR Y	GMA00640
C	IDENT = IDENTITY MATRIX	GMA00650
C	RMN = MEASUREMENT NOISE COVARIANCE MATRIX	GMA00660
C	QPN = PLANT NOISE COVARIANCE MATRIX	GMA00670
C	PNVARX = PLANT NOISE X-SLOPE VARIANCE	GMA00680
C	PNVARY = PLANT NOISE Y-SLOPE VARIANCE	GMA00690
C	PNVARZ = PLANT NOISE Z-SLOPE VARIANCE	GMA00700
C	MNVARX = MEASUREMENT NOISE X-SLOPE VARIANCE	GMA00710
C	MNVARY = MEASUREMENT NOISE Y-SLOPE VARIANCE	GMA00720
C	MNVARZ = MEASUREMENT NOISE Z-SLOPE VARIANCE	GMA00730
C	ISEED = INITIALIZATION FOR RANDOM NUMBER GENERATOR	GMA00740
C	XKAL = X MATRIX	GMA00750
C	Y = OUTPUT MATRIX	GMA00760
C	RNDM = RANDOM NUMBERS USED FOR WHITE NOISE IN MEASUREMENTS AND	GMA00770
C	IN PLANT FORCES	GMA00780
C	BN = B MATRIX TO MULTIPLY NOISE DISTURBANCES	GMA00790
C	TNX,TNY,TNZ= NOISE TORQUES X,Y,Z SLOPES	GMA00800
C	M2=2*MODE	GMA00810
C		GMA00820
C		GMA00830
C		GMA00840
C	***** SAMPLE OF SPACE EXEC FILE *****	GMA00850
C		GMA00860
C	THIS FILE MUST BEGIN IN COLUMN 1 AND RUN WITH THE FOLLOWING	GMA00870
C	SEQUENCE FOR THE INITIAL RUN OF THE PROGRAM:	GMA00880
C		GMA00890
C	FORTVS SPACE (COMPILES PROGRAM)	GMA00900
C	SPACE (EXECUTES EXEC FILE)	GMA00910
C	LOAD SPACE (START (LOADS AND EXECUTES PROGRAM)	GMA00920
C		GMA00930
C	SUBSEQUENT PROGRAM RUNS CAN ELIMINATE "FORTVS SPACE" IF NO	GMA00940
C	CHANGES HAVE BEEN MADE TO THE PROGRAM, AND CAN ELIMINATE	GMA00950
C	RUNNING THE EXEC FILE.	GMA00960
C		GMA00970
C	FI 4 DISK THESIS INPUT B (PERM	GMA00980
C	FI 8 DISK UTILITY DATA (RECFM VS BLOCK 133 PERM	GMA00990
C	FI 11 DISK CNTRL OUTPUT (RECFM F BLOCK 80 LRECL 80 PERM	GMA01000
C	FI 13 DISK GAMMA OUTPUT (RECFM VS BLOCK 133 PERM	GMA01010
C	FI 14 DISK MODE OUTPUT (RECFM F BLOCK 80 LRECL 80 PERM	GMA01020
C	FI 16 DISK COST OUTPUT (RECFM F BLOCK 80 LRECL 80 PERM	GMA01030
C	FI 17 DISK PRT OUTPUT (RECFM F BLOCK 80 LRECL 80 PERM	GMA01040
C	FI 18 DISK ERROR DATA (RECFM F BLOCK 80 LRECL 80 PERM	GMA01050
C	FI 19 DISK END FILE (RECFM F BLOCK 80 LRECL 80 PERM	GMA01060
C	FI 20 DISK GMAT FILE (RECFM F BLOCK 80 LRECL 80 PERM	GMA01070

C		GMA01080
C	*****	GMA01090
C		GMA01100
	PARAMETER (JR=5243, KR=5397, MR=262139)	GMA01110
C		GMA01120
	MIN =1200.0	GMA01130
	WT=1.0D00	GMA01140
	PI = 4.0D0 * ATAN(1.0D0)	GMA01150
C		GMA01160
C	*****	GMA01170
C	***** READ LAMA AND UGVEX MATRICIES *****	GMA01180
C	*****	GMA01190
C		GMA01200
	CALL EXCMS ('CLRSCRN')	GMA01210
	WRITE(6,1008)	GMA01220
	WRITE(6,*) ' READING LAMA AND UGVEX MATRICIES'	GMA01230
	WRITE(6,*) ' '	GMA01240
C	THIS SECTION READS THE LAMA VECTOR AND THE UGVEX	GMA01250
C	MATRIX AND STORES THEM IN MEMORY FOR FURTHER RECALL OF	GMA01260
C	DESIRED LOCATION DATA.	GMA01270
C		GMA01280
	READ(4,1001) NAM	GMA01290
	READ(4,1002)(LAMA(I),I=1,100)	GMA01300
	READ(4,1001) NAM	GMA01310
	DO 5 J = 1,100	GMA01320
	READ(4,1002)(UGVEX(I,J),I=1,684)	GMA01330
5	CONTINUE	GMA01340
C		GMA01350
1001	FORMAT(1X,A6)	GMA01360
1002	FORMAT(1X,8E15.8)	GMA01370
1008	FORMAT(1X,////)	GMA01380
C		GMA01390
500	CALL EXCMS ('CLRSCRN')	GMA01400
C		GMA01410
C	***** STARTING MODE NUMBER *****	GMA01420
C	** SMODE 7 TO 100 (INTEGER) *****	GMA01430
	SMODE=10	GMA01440
C		GMA01450
	WRITE (16,700) SMODE	GMA01460
700	FORMAT (' ', 'STARTING MODE NUMBER: ', I2)	GMA01470
C		GMA01480
C	***** NUMBER OF MODES TO SCAN *****	GMA01490
C	** MODE 1 TO 93 (INTEGER)	GMA01500
C		GMA01510
	MODE= 3	GMA01520
C		GMA01530
	EMODE = SMODE + MODE - 1	GMA01540
C		GMA01550
	WRITE (16,701) MODE	GMA01560
701	FORMAT (' ', 'NUMBER OF MODES SCANNED: ', I2)	GMA01570
C		GMA01580
C	***** NOISE INPUT POSITION *****	GMA01590
C	** NODE 1 TO 114 (INTEGER) (IF 0 THEN NO NOISE INPUT)	GMA01600
	NODE= 8	GMA01610
C		GMA01620

702	WRITE (16,702) NODE	GMA01630
C	FORMAT (' ', 'NOISE NODE LOCATION: ', I5)	GMA01640
C		GMA01650
C	***** SAMPLING TIME *****	GMA01660
C	** SAMPT MUST BE LESS THAN OR EQUAL TO SAMPTM **	GMA01670
	SAMPT = .05	GMA01680
	SAMPTM = ((2.0D0*PI)/SQRT(LAMA(EMODE)))/2.0D0	GMA01690
	IF (SAMPT.GE.SAMPTM) THEN	GMA01700
	SAMPT=SAMPTM	GMA01710
	ENDIF	GMA01720
C		GMA01730
	WRITE (16,900) MIN	GMA01740
900	FORMAT (' ', 2X, 'MIN: ', F8.3)	GMA01750
C		GMA01760
	WRITE (16,703) SAMPT	GMA01770
703	FORMAT (' ', 'SAMPLING TIME: ', D12.4)	GMA01780
C		GMA01790
C	***** DAMPING FACTOR *****	GMA01800
C	** DAMP 0.0 TO 1.0 (REAL*8)	GMA01810
	DAMP=.01	GMA01820
C		GMA01830
	WRITE (16,704) DAMP	GMA01840
704	FORMAT (' ', 'DAMPING FACTOR: ', D12.4)	GMA01850
C		GMA01860
C		GMA01870
C	*** PLANT NOISE VARIANCE ***	GMA01880
C	** PNVARX, PNVARY, PNVARZ GT 0.0	GMA01890
C		GMA01900
	SF1=2.5D06	GMA01910
C		GMA01920
	PNVARX=1.0D00*SF1	GMA01930
	PNVARY=1.0D00*SF1	GMA01940
	PNVARZ=1.0D00*SF1	GMA01950
C		GMA01960
C		GMA01970
C		GMA01980
C		GMA01990
C	*** MEASUREMENT NOISE VARIANCE ***	GMA02000
C	** MNVARX, MNVARY, MNVARZ GT 0.0	GMA02010
	SF=1.0	GMA02020
	MNVX1=5.5D-03*SF	GMA02030
	MNVY1=5.5D-03*SF	GMA02040
	MNVZ1=5.5D-03*SF	GMA02050
C		GMA02060
510	CALL EXCMS ('CLRSCRN')	GMA02070
	WRITE (6,1008)	GMA02080
	WRITE (6,*) ' PROGRAM RUNNING'	GMA02090
C		GMA02100
C	***** NOISE INPUT LOCATION *****	GMA02110
C		GMA02120
	ROWN3 = NODE*6	GMA02130
	ROWN2 = (NODE*6) - 1	GMA02140
	ROWN1 = (NODE*6) - 2	GMA02150
	COUNT = 0	GMA02160
C		GMA02170

C	*****	INITIALIZE MATRICIES	*****	GMA02180
C				GMA02190
	DO 40 I = 1,3			GMA02200
	DO 45 J = 1,3			GMA02210
	RMN(I,J)=0.0			GMA02220
45	CONTINUE			GMA02230
40	CONTINUE			GMA02240
C				GMA02250
	DO 47 I=1,50			GMA02260
	DO 46 J=1,50			GMA02270
	IDENT(I,J)=0.0			GMA02280
	PK(I,J)=0.0			GMA02290
46	CONTINUE			GMA02300
47	CONTINUE			GMA02310
C				GMA02320
	DO 48 K=1,50			GMA02330
	IDENT(K,K)=1.0			GMA02340
48	CONTINUE			GMA02350
C				GMA02360
C	*** INITIALIZE RMN AND QPN MATRICES ***			GMA02370
C				GMA02380
	DO 60 I=1,3			GMA02390
	DO 58 J=1,3			GMA02400
	QPN(I,J)=0.0			GMA02410
58	CONTINUE			GMA02420
60	CONTINUE			GMA02430
C				GMA02440
	RMN(1,1)=MNVX1**2			GMA02450
	RMN(2,2)=MNVY1**2			GMA02460
	RMN(3,3)=MNVZ1**2			GMA02470
	QPN(1,1)=PNVARX**2.0			GMA02480
	QPN(2,2)=PNVARY**2.0			GMA02490
	QPN(3,3)=PNVARZ**2.0			GMA02500
C				GMA02510
9999	FORMAT (' ','')			GMA02520
C	*****			GMA02530
C	*****	BEGIN MAIN PROGRAM	*****	GMA02540
C	*****			GMA02550
C				GMA02560
	CALL STMTRX(EMODE,SMODE,SAMPT,DAMP,PHI,GAMMA,A,B,LAMA,UGVEX,C,			GMA02570
	+ ROWN1,ROWN2,ROWN3,BN)			GMA02580
C				GMA02590
C				GMA02600
C	*** PRE-LOOP PORTION OF KALMAN FILTER			GMA02610
	JK=SMODE*2-2			GMA02620
	M2=2*MODE			GMA02630
	DO 94 I=1,3			GMA02640
	DO 92 J=1,M2			GMA02650
	JL=JK+J			GMA02660
	SUM=0.0			GMA02670
	DO 90 K=1,3			GMA02680
	SUM=SUM+QPN(I,K)*BN(JL,K)			GMA02690
90	CONTINUE			GMA02700
	TMP1(J,I)=SUM			GMA02710
92	CONTINUE			GMA02720
94	CONTINUE			GMA02730

C		GMA02740
C		GMA02750
	DO 98 I=1,M2	GMA02760
	JL=JK+I	GMA02770
	DO 97 J=1,M2	GMA02780
	SUM=0.0	GMA02790
	DO 96 K=1,3	GMA02800
	SUM=SUM+BN(JL,K)*TMP1(J,K)	GMA02810
96	CONTINUE	GMA02820
	BQBT(I,J)=SUM	GMA02830
97	CONTINUE	GMA02840
98	CONTINUE	GMA02850
C		GMA02860
	M2=2*MODE	GMA02870
	DO 100 I=1,M2	GMA02880
	DO 99 J=1,M2	GMA02890
	TMP3(I,J)=0.0	GMA02900
99	CONTINUE	GMA02910
100	CONTINUE	GMA02920
	JL=JK+M2	GMA02930
	DO 9375 I=1,3	GMA02940
	DO 9374 J=1,JL	GMA02950
	C(I,J)=C(I,J)*SF	GMA02960
9374	CONTINUE	GMA02970
9375	CONTINUE	GMA02980
C		GMA02990
C	*****	GMA03000
C	***** THIS SECTION COMPUTES THE STATE UPDATE *****	GMA03010
C	*****	GMA03020
	ESUM=0.0	GMA03030
	COUNT = 0	GMA03040
	ENERGY = 0.0D0	GMA03050
	TIME = 0.0	GMA03060
	CGN=0.0	GMA03070
C	***** SETS LOOP FOR THE ITERATIONS NECESSARY TO OBSERVE *****	GMA03080
C	***** THE SYSTEM FOR THE NUMBER OF MINUTES SPECIFIED *****	GMA03090
C		GMA03100
	LOOP = INT((MIN*60.0)/SAMPT)	GMA03110
	PRT=(DBLE(LOOP))/1200.0	GMA03120
	PRTA=(DBLE(LOOP))/2400.0	GMA03130
	CNTA=0.0	GMA03140
	ACHK=1.0D-10	GMA03150
	H1=0.0	GMA03160
	H2=0.0	GMA03170
	H3=0.0	GMA03180
	H4=0.0	GMA03190
	H5=0.0	GMA03200
	H6=0.0	GMA03210
	TCHK=MIN*60.0	GMA03220
9991	CONTINUE	GMA03230
C		GMA03240
	TIME = TIME+ SAMPT	GMA03250
C		GMA03260
	CGN=CGN+1.0	GMA03270
C		GMA03280
	CNTA=CNTA+1.0	

C	*** START OF KALMAN FILTER ***	GMA03290
C		GMA03300
	M2=2*MODE	GMA03310
C		GMA03320
C	*** COMPUTATION OF PK*AT ***	GMA03330
C		GMA03340
	JK=2*SMODE-2	GMA03350
	DO 175 I=1,M2	GMA03360
	DO 170 J=1,M2	GMA03370
	JL=JK+J	GMA03380
	SUM=0.0	GMA03390
	DO 165 K=1,M2	GMA03400
	JM=JK+K	GMA03410
	SUM =SUM+PK(I,K)*A(JL,JM)	GMA03420
165	CONTINUE	GMA03430
	TMP3(I,J)=SUM	GMA03440
170	CONTINUE	GMA03450
175	CONTINUE	GMA03460
C		GMA03470
C		GMA03480
C	*** COMPUTATION OF A*(PK*AT)+ BQBT = PK1 ***	GMA03490
C		GMA03500
	DO 190 I=1,M2	GMA03510
	JL=JK+I	GMA03520
	DO 185 J=1,M2	GMA03530
	SUM=0.0	GMA03540
	DO 180 K=1,M2	GMA03550
	JM=JK+K	GMA03560
	SUM=SUM+A(JL,JM)*TMP3(K,J)	GMA03570
180	CONTINUE	GMA03580
	PK1(I,J)=SUM+BQBT(I,J)	GMA03590
185	CONTINUE	GMA03600
190	CONTINUE	GMA03610
C	*****	GMA03620
C		GMA03630
C	*** COMPUTE PK1*CT ***	GMA03640
C		GMA03650
	DO 205 I=1,M2	GMA03660
	DO 200 J=1,3	GMA03670
	SUM=0.0	GMA03680
	DO 195 K=1,M2	GMA03690
	JM=JK+K	GMA03700
	SUM=SUM+PK1(I,K)*C(J,JM)	GMA03710
195	CONTINUE	GMA03720
	TMP1(I,J)=SUM	GMA03730
200	CONTINUE	GMA03740
205	CONTINUE	GMA03750
C	*****	GMA03760
C		GMA03770
C	*** COMPUTE C*(PK1*CT)+RMN ***	GMA03780
	DO 220 I=1,3	GMA03790
	DO 215 J=1,3	GMA03800
	SUM=0.0	GMA03810
	DO 210 K=1,M2	GMA03820
	JM=JK+K	GMA03830

	SUM=SUM+C(I,JM)*TMP1(K,J)	GMA03840
210	CONTINUE	GMA03850
	TMP2(I,J)=SUM+RMN(I,J)	GMA03860
215	CONTINUE	GMA03870
220	CONTINUE	GMA03880
C		GMA03890
C	*** COMPUTATION OF THE INVERSE OF C*PK1*CT + R	GMA03900
C		GMA03910
C		GMA03920
	CALL DLINRG (3,TMP2,3,TMP2,3)	GMA03930
C		GMA03940
C	*** COMPUTE CT*INV(C*PK1*CT+R)	GMA03950
C		GMA03960
	DO 245 I=1,M2	GMA03970
	JL=JK+I	GMA03980
	DO 240 J=1,3	GMA03990
	SUM=0.0	GMA04000
	DO 235 K=1,3	GMA04010
	SUM=SUM+C(K,JL)*TMP2(K,J)	GMA04020
235	CONTINUE	GMA04030
	TMP1(I,J)=SUM	GMA04040
240	CONTINUE	GMA04050
245	CONTINUE	GMA04060
C	*****	GMA04070
C		GMA04080
C	*** COMPUTE PK1*C*INV(C*PK1*CT+R) = G ***	GMA04090
C		GMA04100
	DO 260 I=1,M2	GMA04110
	DO 255 J=1,3	GMA04120
	SUM=0.0	GMA04130
	DO 250 K=1,M2	GMA04140
	SUM=SUM+PK1(I,K)*TMP1(K,J)	GMA04150
250	CONTINUE	GMA04160
	G(I,J)=SUM	GMA04170
255	CONTINUE	GMA04180
260	CONTINUE	GMA04190
C		GMA04200
	N9=DABS((G(1,1)-H1)/G(1,1))	GMA04210
	IF (N9.GT.ACHK) THEN	GMA04220
	GO TO 7377	GMA04230
	END IF	GMA04240
	N9=DABS((G(1,3)-H2)/G(1,3))	GMA04250
	IF (N9.GT.ACHK) THEN	GMA04260
	GO TO 7377	GMA04270
	END IF	GMA04280
	N9=DABS((G(2,1)-H3)/G(2,1))	GMA04290
	IF (N9.GT.ACHK) THEN	GMA04300
	GO TO 7377	GMA04310
	END IF	GMA04320
	N9=DABS((G(2,3)-H4)/G(2,3))	GMA04330
	IF (N9.GT.ACHK) THEN	GMA04340
	GO TO 7377	GMA04350
	END IF	GMA04360
	N9=DABS((G(3,3)-H5)/G(3,3))	GMA04370
	IF (N9.GT.ACHK) THEN	GMA04380
	GO TO 7377	GMA04390

END IF	GMA04400
N9=DABS((G(M2,3)-H6)/G(M2,3))	GMA04410
IF (N9.GT.ACHK) THEN	GMA04420
GO TO 7377	GMA04430
END IF	GMA04440
GO TO 400	GMA04450
C	GMA04460
C	GMA04470
7377 CONTINUE	GMA04480
H1=G(1,1)	GMA04490
H2=G(1,3)	GMA04500
H3=G(2,1)	GMA04510
H4=G(2,3)	GMA04520
H5=G(3,3)	GMA04530
H6=G(M2,3)	GMA04540
	GMA04550
	GMA04560
	GMA04570
IF (TCHK.LE.TIME) THEN	GMA04580
GO TO 400	GMA04590
END IF	GMA04600
IF (CGN.GE.PRT) THEN	GMA04610
C	GMA04620
WRITE (6,*) 'TIME= ', TIME, ' SEC.'	GMA04630
C	GMA04640
WRITE (6,*) 'N9= ', N9	GMA04650
CGN=0.0	GMA04660
END IF	GMA04670
C	GMA04680
C	GMA04690
C	GMA04700
**** COMPUTE IDENT - G*C	GMA04710
DO 275 I=1,M2	GMA04720
DO 270 J=1,M2	GMA04730
JL=JK+J	GMA04740
SUM=0.0	GMA04750
DO 265 K=1,3	GMA04760
SUM=SUM+G(I,K)*C(K,JL)	GMA04770
265 CONTINUE	GMA04780
TMP3(I,J)= IDENT(I,J)-SUM	GMA04790
270 CONTINUE	GMA04800
275 CONTINUE	GMA04810
C	GMA04820
C	GMA04830
C	GMA04840
**** COMPUTE PK= (IDENT - G*C)*PK1	GMA04850
C	GMA04860
DO 290 I=1,M2	GMA04870
DO 285 J=1,M2	GMA04880
SUM=0.0	GMA04890
DO 280 K=1,M2	GMA04900
SUM=SUM+TMP3(I,K)*PK1(K,J)	GMA04910
280 CONTINUE	GMA04920
PK(I,J)=SUM	GMA04930
285 CONTINUE	GMA04940
290 CONTINUE	GMA04950
C	
C	

C	END OF KALMAN FILTER PART 1 - START OF PART 2 *****	GMA04960
C		GMA04970
C		GMA04980
C		GMA04990
	GO TO 9991	GMA05000
C		GMA05010
400	CONTINUE	GMA05020
C		GMA05030
	WRITE (20,1008)	GMA05040
	WRITE (20,*) 'TIME= ',TIME	GMA05050
	DO 384 I=1,M2	GMA05060
	WRITE (20,5350) G(I,1),G(I,2),G(I,3)	GMA05070
384	CONTINUE	GMA05080
5350	FORMAT (' ',5X,D15.8 ,5X,D15.8 ,5X,D15.8)	GMA05090
	WRITE (20,*) 'N9= ',N9	GMA05100
C		GMA05110
C	***** COMPUTE AGC = A - G*C	GMA05120
C		GMA05130
	M2=2*MODE	GMA05140
	JK=2*SMODE-2	GMA05150
C		GMA05160
	DO 7155 I=1,M2	GMA05170
	JL=JK+I	GMA05180
	DO 7154 J=1,M2	GMA05190
	JM=JK+J	GMA05200
	SUM=0.0	GMA05210
	DO 7153 K=1,3	GMA05220
	SUM=SUM+G(I,K)*C(K,JM)	GMA05230
7153	CONTINUE	GMA05240
	AGC(I,J)=A(JL,JM)-SUM	GMA05250
7154	CONTINUE	GMA05260
7155	CONTINUE	GMA05270
C		GMA05280
C		GMA05290
C		GMA05300
C	**** COMPUTE THE EIGENVALUES OF AGC	GMA05310
C		GMA05320
	CALL DEVCRG (M2, AGC, 100, EVAL, EVEC, 100)	GMA05330
C		GMA05340
C	***** PRINT EVAL (EIGENVALUE) MATRIX	GMA05350
C		GMA05360
	DO 7157 I=1,M2	GMA05370
	WRITE (20,*) 'I= ', I, 'EIG= ', EVAL(I)	GMA05380
7157	CONTINUE	GMA05390
C		GMA05400
C		GMA05410
C		GMA05420
599	STOP	GMA05430
	END	GMA05440
C		GMA05450
C		GMA05460
C		GMA05470
C		GMA05480
C		GMA05490
C	*****	GMA05500
C	THIS SUBROUTINE COMPUTES THE STATE TRANSITION MATRIX FOR EACH	GMA05510

C	OF THE 100 MODES	GMA05520
C	*****	GMA05530
C		GMA05540
	SUBROUTINE STMTRX(EMODE,SMODE,T,D,PHI,GAMMA,A,B,LAMA,UGVEX,C,	GMA05550
+	ROWN1,ROWN2,ROWN3,BN)	GMA05560
C		GMA05570
	REAL*8 WN,GMA,PHI(2,2,100),GAMMA(2,100),EGT,T,COSW1T,SINW1T	GMA05580
	REAL*8 W1,D,A(200,200),B(200,3),C(6,200),BN(200,3)	GMA05590
	REAL LAMA(100),UGVEX(684,100)	GMA05600
	INTEGER SMODE,R,EMODE,JJ,KK,ROWN1,ROWN2,ROWN3	GMA05610
C		GMA05620
C		GMA05630
C		GMA05640
C		GMA05650
C		GMA05660
	DO 600 I = 1,100	GMA05670
	WN = DBLE(SQRT(LAMA(I)))	GMA05680
	GMA = D*WN/2.0	GMA05690
	EGT = DEXP(-GMA*T)	GMA05700
	W1 = DSQRT((WN**2)-(GMA**2))	GMA05710
	COSW1T = DCOS(W1*T)	GMA05720
	SINW1T = DSIN(W1*T)	GMA05730
C		GMA05740
C		GMA05750
C		GMA05760
C		GMA05770
C		GMA05780
	IF(WN.EQ.0)THEN	GMA05790
	PHI(1,1,I) = EGT*COSW1T	GMA05800
	PHI(1,2,I) = T	GMA05810
	PHI(2,1,I) = 0	GMA05820
	PHI(2,2,I) = EGT*COSW1T	GMA05830
C		GMA05840
C		GMA05850
C		GMA05860
C		GMA05870
C		GMA05880
C		GMA05890
	GAMMA(1,I) = 0	GMA05900
	GAMMA(2,I) = 0	GMA05910
	ELSE	GMA05920
C		GMA05930
C		GMA05940
C		GMA05950
C		GMA05960
C		GMA05970
	PHI(1,1,I) = EGT*(COSW1T + (GMA*(W1**(-1)))*SINW1T)	GMA05980
	PHI(1,2,I) = (W1**(-1))*EGT*SINW1T	GMA05990
	PHI(2,1,I) = -(WN**2)*(W1**(-1))*EGT*SINW1T	GMA06000
	PHI(2,2,I) = EGT*(COSW1T - (GMA*(W1**(-1)))*SINW1T)	GMA06010
C		GMA06020
C		GMA06030
C		GMA06040
C		GMA06050
	GAMMA(1,I)=(WN**(-2))*(1.D0-EGT*COSW1T-EGT*(GMA/W1)*SINW1T)	GMA06060

C	GAMMA(2,I) = (W1**(-1))*EGT*SINW1T	GMA06070
C		GMA06080
C		GMA06090
C		GMA06100
	ENDIF	GMA06110
C		GMA06120
C		GMA06130
C		GMA06140
600	CONTINUE	GMA06150
C		GMA06160
C		GMA06170
C		GMA06180
C		GMA06190
C	R = 1	GMA06200
C		GMA06210
	DO 610 K = 1 ,100	GMA06220
C		GMA06230
C		GMA06240
C		GMA06250
C		GMA06260
C		GMA06270
C		GMA06280
	A(R,R) = PHI(1,1,K)	GMA06290
	A(R,R+1) = PHI(1,2,K)	GMA06300
	A(R+1,R) = PHI(2,1,K)	GMA06310
	A(R+1,R+1) = PHI(2,2,K)	GMA06320
C		GMA06330
C		GMA06340
C		GMA06350
C		GMA06360
C		GMA06370
C	*** B MATRIX FOR MULTIPLYING CONTROL TORQUES	GMA06380
C		GMA06390
	B(R,1) = GAMMA(1,K)*DBLE(UGVEX(412,K))	GMA06400
	B(R,2) = GAMMA(1,K)*DBLE(UGVEX(413,K))	GMA06410
	B(R,3) = GAMMA(1,K)*DBLE(UGVEX(414,K))	GMA06420
	B(R+1,1) = GAMMA(2,K)*DBLE(UGVEX(412,K))	GMA06430
	B(R+1,2) = GAMMA(2,K)*DBLE(UGVEX(413,K))	GMA06440
	B(R+1,3) = GAMMA(2,K)*DBLE(UGVEX(414,K))	GMA06450
C		GMA06460
C		GMA06470
C		GMA06480
C		GMA06490
C		GMA06500
C		GMA06510
C		GMA06520
C	*** BN MATRIX FOR MULTIPLYING THE NOISE DISTURBANCES	GMA06530
C		GMA06540
C		GMA06550
C		GMA06560
C		GMA06570
	BN(R,1)=GAMMA(1,K)*DBLE(UGVEX(ROWN1,K))	GMA06580
	BN(R,2)=GAMMA(1,K)*DBLE(UGVEX(ROWN2,K))	GMA06590
	BN(R,3)=GAMMA(1,K)*DBLE(UGVEX(ROWN3,K))	GMA06600
	BN(R+1,1)=GAMMA(2,K)*DBLE(UGVEX(ROWN1,K))	GMA06610
	BN(R+1,2)=GAMMA(2,K)*DBLE(UGVEX(ROWN2,K))	GMA06620

	BN(R+1,3)=GAMMA(2,K)*DBLE(UGVEX(ROWN3,K))	GMA06630
C		GMA06640
C		GMA06650
C		GMA06660
C		GMA06670
C		GMA06680
C		GMA06690
	R = R+2	GMA06700
610	CONTINUE	GMA06710
C		GMA06720
C		GMA06730
C		GMA06740
C		GMA06750
C		GMA06760
C		GMA06770
C		GMA06780
C	*** C MATRIX PRODUCTION ***	GMA06790
C		GMA06800
C		GMA06810
C		GMA06820
	JJ=-1	GMA06830
	DO 640 I=1,100	GMA06840
	JJ=JJ+1	GMA06850
	KK=I+JJ	GMA06860
C		GMA06870
C		GMA06880
	C(1,KK) = DBLE(UGVEX(418,I))	GMA06890
	C(2,KK) = DBLE(UGVEX(419,I))	GMA06900
	C(3,KK) = DBLE(UGVEX(420,I))	GMA06910
C		GMA06920
C		GMA06930
C		GMA06940
	KK=KK+1	GMA06950
C		GMA06960
	C(1,KK)=0.0	GMA06970
	C(2,KK)=0.0	GMA06980
	C(3,KK)=0.0	GMA06990
C		GMA07000
640	CONTINUE	GMA07010
C		GMA07020
C		GMA07030
C		GMA07040
	RETURN	GMA07050
	END	GMA07060

APPENDIX B. KALMAN OBSERVER AND PLANT SIMULATION

C		SIM00010
C	*****	SIM00020
C	***** SIMRUN *****	SIM00030
C	***** ADAPTED TO READ KALMAN FILETER G MATRICE *****	SIM00040
C	***** THEN RUN ALL N MODES OF THE PLANT WHILE *****	SIM00050
C	***** USING A KALMAN FILTER TO OBSERVE M *****	SIM00060
C	***** NUMBER OF STATES *****	SIM00070
C	*****	SIM00080
C		SIM00090
C		SIM00100
C	*****	SIM00110
C	***** VARIABLE DECLARATIONS *****	SIM00120
C	*****	SIM00130
C		SIM00140
	EXTERNAL STMTRX,EXCMS	SIM00150
	CHARACTER*6 NAM	SIM00160
	CHARACTER*1 AGAIN,CORECT,RAGAIN	SIM00170
	INTEGER ROWN1,ROWN2,ROWN3,COUNT,NODE,MODE,KQ,EMODE,SMODE,R2M,C2M	SIM00180
	INTEGER CT,CF,KADJ,CFADJ,LOOP,PRNT,JJ,JK,N1,JR,KR,MR,ISEED,M2	SIM00190
	INTEGER ITYPE(200), IPVT(100), NS, NF, SN, FN	SIM00200
	INTEGER JL,J1,JM , JP, JQ, KA, KB, KC, KD, KE, KF, KG	SIM00210
C		SIM00220
C		SIM00230
C		SIM00240
	REAL LAMA(100), UGVEX(684,100),RNODE,RMODE,MIN	SIM00250
	REAL*8 PHI(2,2,100),GAMMA(2,100),EGT,GMA,WN,W1,X1T,X2T,TIME	SIM00260
	REAL*8 A(200,200),B(200,3),F(3, 50),IMPLSE,ENERGY	SIM00270
	REAL*8 COSW1T,SINW1T,X(200)	SIM00280
	REAL*8 TCX,TCY,TCZ,DAMP,SAMPT,PI,SAMPTM,SUM1,SUM2,SUM3,SUMC	SIM00290
	REAL*8 C(3,200), IDENT(50, 50), RMN(3,3), QPN(3,3)	SIM00300
	REAL*8 Y(3) , BN(200,3)	SIM00310
	REAL*8 PNVARX, PNVARY, PNVARZ	SIM00320
	REAL*8 MNVARX, MNVARY, MNVARZ	SIM00330
	REAL*8 SUM, RNDM(6), RND1, RND2	SIM00340
	REAL*8 XH(50) ,BQBT(50, 50)	SIM00350
	REAL*8 SF1	SIM00360
	REAL*8 TMP1(50,3), TMP2(3,3), TMP3(50, 50)	SIM00370
	REAL*8 G(50,3)	SIM00380
	REAL*8 XH1(50) ,DY(3) , ES,ED,ESUM,CGN,PRT	SIM00390
	REAL*8 WT , WD(3), BNWD(200)	SIM00400
	REAL*8 AX(200) , V(3), SF , TO, CTT, ESS	SIM00410
	REAL*8 CTG, XDEL, E2(100), XDEL1, ERS, PRT1, E3(100), XS(100)	SIM00420
C		SIM00430
C	*****	SIM00440
C	***** VARIABLE DEFINITIONS *****	SIM00450
C	*****	SIM00460
C		SIM00470
C	STMTRX = SUBROUTINE EXTABLISHES STATE TRANSITION MATRICIES	SIM00480
C	LAMA = VECTOR OF THE SQUARE OF THE NATURAL FREQUENCIES	SIM00490
C	UGVEX = MODE POSITONS AND SLOPES OF THE NODAL POINTS	SIM00500
C	PHI = STATE TRANSITION MATRICIES FOR EACH MODE	SIM00510

C	GAMMA = INPUT TRANSITION MATRIX	SIM00520
C	A = DIAGONAL MATRIX CONSISTING OF PHI	SIM00530
C	B = INPUT MATRIX OF GAMMA AND CONTROL SLOPES	SIM00540
C	DAMP = DAMPING FACTOR	SIM00550
C	SAMPT = SAMPLING TIME	SIM00560
C	TCX, TCY, TCZ = CONTROL TORQUE VALUES	SIM00570
C	ENERGY = TOTAL SYSTEM ENERGY	SIM00580
C	IMPLSE = IMPULSE INPUT FUNCTION	SIM00590
C	MIN = NUMBER OF MINUTES SYSTEM WILL BE OBSERVED	SIM00600
C	SMODE = NUMBER OF STARTING MODE (INT)	SIM00610
C	MODE = NUMBER OF MODES (INT)	SIM00620
C	EMODE = NUMBER OF THE LAST MODE (INT)	SIM00630
C	NODE = NUMBER OF THE NOISE INPUT MODE (INT)	SIM00640
C	*** NOISE SLOPE LOCATIONS IN DATA MATRIX ***	SIM00650
C	ROWN1 = X-SLOPE LOCATION	SIM00660
C	ROWN2 = Y-SLOPE LOCATION	SIM00670
C	ROWN3 = Z-SLOPE LOCATION	SIM00680
C	C = OUTPUT MATRIX FOR Y	SIM00690
C	IDENT = IDENTITY MATRIX	SIM00700
C	RMN = MEASUREMENT NOISE COVARIANCE MATRIX	SIM00710
C	QPN = PLANT NOISE COVARIANCE MATRIX	SIM00720
C	PNVARX = PLANT NOISE X-SLOPE VARIANCE	SIM00730
C	PNVARY = PLANT NOISE Y-SLOPE VARIANCE	SIM00740
C	PNVARZ = PLANT NOISE Z-SLOPE VARIANCE	SIM00750
C	MNVARX = MEASUREMENT NOISE X-SLOPE VARIANCE	SIM00760
C	MNVARY = MEASUREMENT NOISE Y-SLOPE VARIANCE	SIM00770
C	MNVARZ = MEASUREMENT NOISE Z-SLOPE VARIANCE	SIM00780
C	ISEED = INITIALIZATION FOR RANDOM NUMBER GENERATOR	SIM00790
C	XKAL = X MATRIX	SIM00800
C	Y = OUTPUT MATRIX	SIM00810
C	RNDM = RANDOM NUMBERS USED FOR WHITE NOISE IN MEASUREMENTS AND	SIM00820
C	IN PLANT FORCES	SIM00830
C	BN = B MATRIX TO MULTIPLY NOISE DISTURBANCES	SIM00840
C	TNX,TNY,TNZ= NOISE TORQUES X,Y,Z SLOPES	SIM00850
C	M2=2*MODE	SIM00860
C		SIM00870
C		SIM00880
C		SIM00890
C	***** SAMPLE OF SPACE EXEC FILE *****	SIM00900
C		SIM00910
C	THIS FILE MUST BEGIN IN COLUMN 1 AND RUN WITH THE FOLLOWING	SIM00920
C	SEQUENCE FOR THE INITIAL RUN OF THE PROGRAM:	SIM00930
C		SIM00940
C	FORTVS SPACE (COMPILES PROGRAM)	SIM00950
C	SPACE (EXECUTES EXEC FILE)	SIM00960
C	LOAD SPACE (START (LOADS AND EXECUTES PROGRAM)	SIM00970
C		SIM00980
C	SUBSEQUENT PROGRAM RUNS CAN ELIMINATE "FORTVS SPACE" IF NO	SIM00990
C	CHANGES HAVE BEEN MADE TO THE PROGRAM, AND CAN ELIMINATE	SIM01000
C	RUNNING THE EXEC FILE.	SIM01010
C		SIM01020
C	FI 4 DISK THESIS INPUT B (PERM	SIM01030
C	FI 8 DISK UTILITY DATA (RECFM VS BLOCK 133 PERM	SIM01040
C	FI 11 DISK CNTRL OUTPUT (RECFM F BLOCK 80 LRECL 80 PERM	SIM01050
C	FI 13 DISK GAMMA OUTPUT (RECFM VS BLOCK 133 PERM	SIM01060
C	FI 14 DISK MODE OUTPUT (RECFM F BLOCK 80 LRECL 80 PERM	SIM01070

C	FI 16 DISK COST OUTPUT (RECFM F BLOCK 80 LRECL 80 PERM	SIM01080
C	FI 17 DISK PRT OUTPUT (RECFM F BLOCK 80 LRECL 80 PERM	SIM01090
C	FI 18 DISK ERROR DATA (RECFM F BLOCK 80 LRECL 80 PERM	SIM01100
C	FI 19 DISK END FILE (RECFM F BLOCK 80 LRECL 80 PERM	SIM01110
C	FI 20 DISK GMAT FILE (RECFM F BLOCK 80 LRECL 80 PERM	SIM01120
C		SIM01130
C	*****	SIM01140
C	PARAMETER (JR=5243, KR=5397, MR=262139)	SIM01150
C		SIM01160
C		SIM01170
C		SIM01180
C	MIN =1.00	SIM01190
C		SIM01200
C	WT=1.0D00	SIM01210
C	PI = 4.0D0 * ATAN(1.0D0)	SIM01220
C		SIM01230
C	*****	SIM01240
C	***** READ LAMA AND UGVEX MATRICIES *****	SIM01250
C	*****	SIM01260
C		SIM01270
C	CALL EXCMS ('CLRSCRN')	SIM01280
C	WRITE(6,1008)	SIM01290
C	WRITE(6,*) ' READING LAMA AND UGVEX MATRICIES'	SIM01300
C	WRITE(6,*) ' '	SIM01310
C	THIS SECTION READS THE LAMA VECTOR AND THE UGVEX	SIM01320
C	MATRIX AND STORES THEM IN MEMORY FOR FURTHER RECALL OF	SIM01330
C	DESIRED LOCATION DATA.	SIM01340
C		SIM01350
C	READ(4,1001) NAM	SIM01360
C	READ(4,1002)(LAMA(I),I=1,100)	SIM01370
C	READ(4,1001) NAM	SIM01380
C	DO 5 J = 1,100	SIM01390
C	READ(4,1002)(UGVEX(I,J),I=1,684)	SIM01400
5	CONTINUE	SIM01410
C		SIM01420
1001	FORMAT(1X,A6)	SIM01430
1002	FORMAT(1X,8E15.8)	SIM01440
1008	FORMAT (1X,////)	SIM01450
C		SIM01460
500	CALL EXCMS ('CLRSCRN')	SIM01470
C		SIM01480
C	***** STARTING MODE NUMBER *****	SIM01490
C	** SMODE 7 TO 100 (INTEGER) ****	SIM01500
C	SMODE= 7	SIM01510
C		SIM01520
C	WRITE (16,700) SMODE	SIM01530
700	FORMAT (' ', 'STARTING MODE NUMBER: ', I2)	SIM01540
C		SIM01550
C	***** NUMBER OF MODES TO SCAN *****	SIM01560
C	** MODE 1 TO 93 (INTEGER)	SIM01570
C		SIM01580
C	MODE=20	SIM01590
C		SIM01600
C	EMODE = SMODE + MODE - 1	SIM01610
C		SIM01620

	WRITE (16,701) MODE	SIM01630
701	FORMAT (' ', 'NUMBER OF MODES SCANNED: ', I2)	SIM01640
C		SIM01650
C	***** NOISE INPUT POSITION *****	SIM01660
C	** NODE 1 TO 114 (INTEGER) (IF 0 THEN NO NOISE INPUT)	SIM01670
	NODE= 8	SIM01680
C		SIM01690
	WRITE (16,702) NODE	SIM01700
702	FORMAT (' ', ' NOISE NODE LOCATION: ', I5)	SIM01710
C		SIM01720
C	***** START AND STOP FOR PLANT	SIM01730
	SN=7	SIM01740
	FN=20	SIM01750
	NS=SN*2-1	SIM01760
	NF=SN*2+2*FN-2	SIM01770
	WRITE (16,899) SN, FN	SIM01780
899	FORMAT (' ', 'PLANT -- SN= ', I5, ' FN= ', I5)	SIM01790
C	***** SAMPLING TIME *****	SIM01800
C	** SAMPT MUST BE LESS THAN OR EQUAL TO SAMPTM **	SIM01810
	SAMPT = 0.05	SIM01820
	SAMPTM = ((2.0D0*PI)/SQRT(LAMA(EMODE)))/1.0D01	SIM01830
	IF (SAMPT.GE. SAMPTM) THEN	SIM01840
	SAMPT=SAMPTM	SIM01850
	ENDIF	SIM01860
C		SIM01870
	WRITE (16,900) MIN	SIM01880
900	FORMAT (' ', 2X, 'MIN: ', F8.3)	SIM01890
C		SIM01900
	WRITE (16,703) SAMPT, SAMPTM	SIM01910
703	FORMAT (' ', 'SAMPLING TIME: ', D12.4, 2X, 'SAMPTM= ', D15.8)	SIM01920
C		SIM01930
C	***** DAMPING FACTOR *****	SIM01940
C	** DAMP 0.0 TO 1.0 (REAL*8)	SIM01950
	DAMP=.01	SIM01960
C		SIM01970
	WRITE (16,704) DAMP	SIM01980
704	FORMAT (' ', 'DAMPING FACTOR: ', D12.4)	SIM01990
C		SIM02000
C		SIM02010
C	*** PLANT NOISE VARIANCE ***	SIM02020
C	** PNVARX, PNVAR Y, PNVARZ GT 0.0	SIM02030
	SF1=2.5D06	SIM02040
	SF=1.0D00	SIM02050
C		SIM02060
	PNVARX=1.0D00*SF1	SIM02070
	PNVAR Y=1.0D00*SF1	SIM02080
	PNVARZ=1.0D00*SF1	SIM02090
C		SIM02100
C		SIM02110
C		SIM02120
C	*** MEASUREMENT NOISE VARIANCE ***	SIM02130
C	** MNVARX, MNVAR Y, MNVARZ GT 0.0	SIM02140
	MNVARX=1.0D-03 *SF	SIM02150
	MNVAR Y=1.0D-03 *SF	SIM02160

	MNVARZ=1.0D-03 *SF	SIM02170
C		SIM02180
C		SIM02190
	WRITE (16,711)	SIM02200
711	FORMAT(' ','PLANT NOISE VARIANCE: ')	SIM02210
	WRITE (16,712)	SIM02220
712	FORMAT(' ',6X,'PNVARX',13X,'PNVARY',13X,'PNVARZ')	SIM02230
	WRITE (16,713) PNVARX, PNVARY, PNVARZ	SIM02240
713	FORMAT(' ',2X,E15.8,2X,E15.8,2X,E15.8)	SIM02250
	WRITE(16,714)	SIM02260
714	FORMAT(' ','MEASUREMENT NOISE: ')	SIM02270
	WRITE(16,715)	SIM02280
715	FORMAT(' ',6X,'MNVARX',13X,'MNVARY',13X,'MNVARZ')	SIM02290
	WRITE(16,713) MNVARX,MNVARY,MNVARZ	SIM02300
C		SIM02310
510	CALL EXCMS ('CLRSCRN')	SIM02320
	WRITE (6,1008)	SIM02330
	WRITE (6,*) 'PROGRAM RUNNING'	SIM02340
C		SIM02350
C	***** NOISE INPUT LOCATION *****	SIM02360
C		SIM02370
	ROWN3 = NODE*6	SIM02380
	ROWN2 = (NODE*6) - 1	SIM02390
	ROWN1 = (NODE*6) - 2	SIM02400
	COUNT = 0	SIM02410
C		SIM02420
C		SIM02430
C	***** INITIALIZE MATRICIES *****	SIM02440
C		SIM02450
	DO 48 K=1,50	SIM02460
	IDENT(K,K)=1.0	SIM02470
48	CONTINUE	SIM02480
C		SIM02490
	DO 54 K = 1, 200	SIM02500
	X(K) = 0.0	SIM02510
54	CONTINUE	SIM02520
C		SIM02530
C		SIM02540
	WRITE(6,1008)	SIM02550
	WRITE (6,*) ' INITIALIZE RMN AND QPN MATRICES '	SIM02560
C	**** INITIALIZE RMN AND QPN MATRICES ****	SIM02570
C		SIM02580
	DO 60 I=1,3	SIM02590
	DO 58 J=1,3	SIM02600
	RMN(I,J)=0.0	SIM02610
	QPN(I,J)=0.0	SIM02620
58	CONTINUE	SIM02630
60	CONTINUE	SIM02640
C		SIM02650
	RMN(1,1)=MNVARX**2	SIM02660
	RMN(2,2)=MNVARY**2	SIM02670
	RMN(3,3)=MNVARZ**2	SIM02680
	QPN(1,1)=PNVARX**2	SIM02690
	QPN(2,2)=PNVARY**2	SIM02700

	QPN(3,3)=PNVARZ**2.0	SIM02710
C		SIM02720
C		SIM02730
	WRITE(6,1008)	SIM02740
	WRITE(6,*) ' ENTER STMTRX '	SIM02750
C		SIM02760
C	*****	SIM02770
C	***** BEGIN MAIN PROGRAM *****	SIM02780
C	*****	SIM02790
C		SIM02800
	CALL STMTRX(EMODE,SMODE,SAMPT,DAMP,PHI,GAMMA,A,B,LAMA,UGVEX,C,	SIM02810
	+ ROWN1,ROWN2,ROWN3,BN)	SIM02820
C		SIM02830
C		SIM02840
	WRITE (16,1008)	SIM02850
	DO 11000 I=1,200	SIM02860
	DO 10900 J=1,3	SIM02870
	C(J,I)= C(J,I)*SF	SIM02880
10900	CONTINUE	SIM02890
11000	CONTINUE	SIM02900
C		SIM02910
C	*** PRE-LOOP PORTION OF KALMAN FILTER	SIM02920
C		SIM02930
	M2=2*MODE	SIM02940
	JP=2*SMODE-1	SIM02950
	JQ=2*EMODE	SIM02960
	DO 90 I=1,50	SIM02970
	XH(I)=0.0	SIM02980
90	CONTINUE	SIM02990
C		SIM03000
	DO 9971 I=1,M2	SIM03010
	READ (20,*) G(I,1), G(I,2), G(I,3)	SIM03020
9971	CONTINUE	SIM03030
C		SIM03040
C		SIM03050
	WRITE (14,1008)	SIM03060
	DO 384 I=1,M2	SIM03070
	WRITE (14,5350) G(I,1),G(I,2),G(I,3)	SIM03080
384	CONTINUE	SIM03090
5350	FORMAT (' ',2X,D15.8,2X,D15.8,2X,D15.8)	SIM03100
C		SIM03110
C		SIM03120
C	*****	SIM03130
C	***** THIS SECTION COMPUTES THE STATE UPDATE *****	SIM03140
C	*****	SIM03150
	DO 9771 I=1,100	SIM03160
	E2(I)=0.0	SIM03170
	E3(I)=0.0	SIM03180
	XS(I)=0.0	SIM03190
9771	CONTINUE	SIM03200
	ESS =0.0	SIM03210
	COUNT = 0	SIM03220
	ENERGY = 0.0D0	SIM03230
	TIME = 0.0	SIM03240
	CGN=0.0	SIM03250

	CTG=0.0	SIM03260
C	***** SETS LOOP FOR THE ITERATIONS NECESSARY TO OBSERVE	SIM03270
C	***** THE SYSTEM FOR THE NUMBER OF MINUTES SPECIFIED	SIM03280
	WRITE (6,1008)	SIM03290
	WRITE (6,*) ' START STATE UPDATE '	SIM03300
	LOOP = INT((MIN*60.0)/SAMPT)	SIM03310
	PRT= (DBLE(LOOP))/30.0	SIM03320
	CTT=0.0	SIM03330
C		SIM03340
	DO 400 L = 0, LOOP	SIM03350
	TIME = DBLE(L)*SAMPT	SIM03360
C		SIM03370
	IF(L.EQ.0)THEN	SIM03380
	IMPLSE =(1.0D06*SF1)/(DSQRT(SAMPT))	SIM03390
	ELSE	SIM03400
	IMPLSE = 0.0D0	SIM03410
	ENDIF	SIM03420
C		SIM03430
	TO=0.0	SIM03440
C	***** RANDOM NUMBER GENERATOR *****	SIM03450
C		SIM03460
	DO 101 I=1,6	SIM03470
	ISEED=MOD(ISEED*JR+KR,MR)	SIM03480
	RND1=(DBLE(ISEED)+0.5D00)/DBLE(MR)	SIM03490
	ISEED=MOD(ISEED*JR+KR,MR)	SIM03500
	RND2=(DBLE(ISEED)+0.5D00)/DBLE(MR)	SIM03510
	RNDM(I)=DSQRT(-2.0*DLOG(RND1))*DCOS(6.2831853D00*RND2)	SIM03520
101	CONTINUE	SIM03530
C	*****	SIM03540
	CTT=CTT+1.0	SIM03550
C	***** START OF STATE UPDATE ****	SIM03560
C		SIM03570
C	*** COMPUTE $AX^{0200} = A^{0200} X^{0200} * X^{0200}$	SIM03580
C		SIM03590
C		SIM03600
C	*** COMPUTE $AX = A * X$	SIM03610
C		SIM03620
	JK=SMODE*2-2	SIM03630
	JP=JK+1	SIM03640
	JQ=2*EMODE	SIM03650
C		SIM03660
C		SIM03670
	DO 5015 I=NS,NF	SIM03680
	SUM=0.0	SIM03690
	DO 5010 K=NS,NF	SIM03700
	SUM=SUM+A(I,K)*X(K)	SIM03710
5010	CONTINUE	SIM03720
	AX(I)=SUM	SIM03730
5015	CONTINUE	SIM03740
C		SIM03750
C	*** COMPUTE WD^{03}	SIM03760
C		SIM03770
	WD(1)=PNVARX*RNDM(1)*TO+IMPLSE	SIM03780
	WD(2)=PNVARY*RNDM(2)*TO	SIM03790
	WD(3)=PNVARZ*RNDM(3)*TO	SIM03800
C		SIM03810

C		SIM03820
C	**** COMPUTE BNWD ⁰ 200 =BN ⁰ 200 X 3 * WD ⁰ 3	SIM03830
C		SIM03840
	DO 5035 I=NS,NF	SIM03850
	SUM=0.0	SIM03860
	DO 5030 K=1,3	SIM03870
	SUM=SUM+BN(I,K)*WD(K)	SIM03880
5030	CONTINUE	SIM03890
	BNWD(I)=SUM	SIM03900
5035	CONTINUE	SIM03910
C		SIM03920
C	**** COMPUTE X ⁰ 200 =AX ⁰ 200 + BNWD ⁰ 200	SIM03930
C		SIM03940
	DO 5040 I=NS,NF	SIM03950
	X(I)= AX(I) + BNWD(I)	SIM03960
	IF (DABS(X(I)).LT. 1.0D-60) THEN	SIM03970
	X(I)=1.0D-60	SIM03980
	END IF	SIM03990
C		SIM04000
C		SIM04010
C		SIM04020
5040	CONTINUE	SIM04030
C		SIM04040
C	**** COMPUTE V ⁰ 3	SIM04050
C		SIM04060
	V(1)=MNVARX*RNDM(4)	SIM04070
	V(2)=MNVARY*RNDM(5)	SIM04080
	V(3)=MNVARZ*RNDM(6)	SIM04090
C		SIM04100
C	**** COMPUTE Y ⁰ 3 = C ⁰ 3 X 200 * X ⁰ 200 + V ⁰ 3	SIM04110
C		SIM04120
	DO 5050 I=1,3	SIM04130
	SUM=0.0	SIM04140
	DO 5045 K=NS,NF	SIM04150
	SUM=SUM+C(I,K)*X(K)	SIM04160
5045	CONTINUE	SIM04170
	Y(I)=SUM+V(I)	SIM04180
5050	CONTINUE	SIM04190
C		SIM04200
C	*****	SIM04210
C		SIM04220
C	**** START OF KALMAN FILTER ****	SIM04230
C		SIM04240
	M2=2*MODE	SIM04250
C		SIM04260
C		SIM04270
C	**** COMPUTE XH1 = A*XH	SIM04280
C		SIM04290
	DO 300 I=JP,JQ	SIM04300
	SUM=0.0	SIM04310
	DO 295 K=JP,JQ	SIM04320
	SUM=SUM+A(I,K) * XH(K)	SIM04330
295	CONTINUE	SIM04340
	XH1(I)=SUM	SIM04350
300	CONTINUE	SIM04360
C	*****	SIM04370

C		SIM04380
C	*** COMPUTE DY = Y - C*XH1	SIM04390
C		SIM04400
	DO 315 I=1,3	SIM04410
	SUM=0.0	SIM04420
	DO 310 K=JP,JQ	SIM04430
	SUM=SUM+C(I,K)*XH1(K)	SIM04440
310	CONTINUE	SIM04450
	DY(I)=Y(I)-SUM	SIM04460
315	CONTINUE	SIM04470
C		SIM04480
C	*****	SIM04490
C		SIM04500
C	*** COMPUTE XH = XH1 + G*DY	SIM04510
C		SIM04520
	DO 325 I=1,M2	SIM04530
	J1=JP-1+I	SIM04540
	SUM=0.0	SIM04550
	DO 320 K=1,3	SIM04560
	SUM=SUM+G(I,K)*DY(K)	SIM04570
320	CONTINUE	SIM04580
	XH(J1)=XH1(J1)+SUM	SIM04590
	IF (DABS(XH(J1)).LT.1.0D-60) THEN	SIM04600
	XH(J1)=1.0*D-60	SIM04610
	END IF	SIM04620
C		SIM04630
325	CONTINUE	SIM04640
C		SIM04650
C		SIM04660
C		SIM04670
C	***** END OF KALMAN ROUTINES *****	SIM04680
C		SIM04690
C	*** COMPUTATION OF ESUM ***	SIM04700
	DO 340 I=JP,JQ	SIM04710
	XDEL= X(I)-XH(I)	SIM04720
	XDEL1=XDEL*XDEL*SAMPT	SIM04730
	E2(I)=E2(I)+XDEL1	SIM04740
	XS(I)=XS(I)+X(I)*X(I)*SAMPT	SIM04750
	E3(I)=E2(I)/XS(I)	SIM04760
340	CONTINUE	SIM04770
C		SIM04780
	CGN=CGN+1.0	SIM04790
	IF (CTT.EQ.1.0.OR.CGN.GT.PRT) THEN	SIM04800
C		SIM04810
	WRITE (6,*) 'TIME= ', TIME, ' SEC.'	SIM04820
C		SIM04830
	WRITE (17,1008)	SIM04840
	WRITE (16,1008)	SIM04850
	WRITE (16,2100) TIME	SIM04860
C		SIM04870
	WRITE (17,2100) TIME	SIM04880
2100	FORMAT(' ', 'TIME= ', F9.3)	SIM04890
	DO 380 I=JP , JQ	SIM04900
	WRITE (16,4500) I,X(I) ,I ,XH(I)	SIM04910
C		SIM04920

380	WRITE (17,4530) I,E2(I) ,E3(I) , XS(I)	SIM04930
	CONTINUE	SIM04940
C		SIM04950
C		SIM04960
C		SIM04970
C		SIM04980
	CGN=0.0	SIM04990
	END IF	SIM05000
4500	FORMAT (' ', ' X(' ,I3,')= ',D15.8,2X,'XH(' ,I3,')= ',D15.8)	SIM05010
4530	FORMAT (' ',5X,I5,5X,3 D15.8)	SIM05020
C		SIM05030
400	CONTINUE	SIM05040
C		SIM05050
C		SIM05060
	DO 401 I=JP,JQ	SIM05070
	WRITE (19,4530) I, E2(I) ,E3(I), XS(I)	SIM05080
401	CONTINUE	SIM05090
C		SIM05100
C		SIM05110
C		SIM05120
C		SIM05130
599	STOP	SIM05140
	END	SIM05150
C		SIM05160
C		SIM05170
C	*****	SIM05180
C	THIS SUBROUTINE COMPUTES THE STATE TRANSITION MATRIX FOR EACH	SIM05190
C	OF THE 100 MODES	SIM05200
C	*****	SIM05210
C		SIM05220
	SUBROUTINE STMTRX(EMODE,SMODE,T,D,PHI,GAMMA,A,B,LAMA,UGVEX,C,	SIM05230
+	ROWN1,ROWN2,ROWN3,BN)	SIM05240
C		SIM05250
	REAL*8 WN,GMA,PHI(2,2,100),GAMMA(2,100),EGT,T,COSW1T,SINW1T	SIM05260
	REAL*8 W1,D,A(200,200),B(200,3),C(3,200),BN(200,3)	SIM05270
	REAL LAMA(100),UGVEX(684,100)	SIM05280
	INTEGER SMODE,R,EMODE,JJ,KK,ROWN1,ROWN2,ROWN3	SIM05290
C		SIM05300
C		SIM05310
	DO 600 I = 1 ,100	SIM05320
	WN = DBLE(SQRT(LAMA(I)))	SIM05330
	GMA = D*WN/2.0	SIM05340
	EGT = DEXP(-GMA*T)	SIM05350
	W1 = DSQRT((WN**2)-(GMA**2))	SIM05360
	COSW1T = DCOS(W1*T)	SIM05370
	SINW1T = DSIN(W1*T)	SIM05380
C		SIM05390
C		SIM05400
C		SIM05410
C		SIM05420
	IF(WN.EQ.0)THEN	SIM05430
	PHI(1,1,I) = EGT*COSW1T	SIM05440
	PHI(1,2,I) = T	SIM05450
	PHI(2,1,I) = 0	SIM05460
	PHI(2,2,I) = EGT*COSW1T	SIM05470
C		SIM05480

C		SIM05490
C		SIM05500
C		SIM05510
C		SIM05520
C		SIM05530
	GAMMA(1,I) = 0	SIM05540
	GAMMA(2,I) = 0	SIM05550
	ELSE	SIM05560
C		SIM05570
C		SIM05580
C		SIM05590
C		SIM05600
	PHI(1,1,I) = EGT*(COSW1T + (GMA*(W1**(-1)))*SINW1T)	SIM05610
	PHI(1,2,I) = (W1**(-1))*EGT*SINW1T	SIM05620
	PHI(2,1,I) = -(WN**2)*(W1**(-1))*EGT*SINW1T	SIM05630
	PHI(2,2,I) = EGT*(COSW1T - (GMA*(W1**(-1)))*SINW1T)	SIM05640
C		SIM05650
C		SIM05660
C		SIM05670
C		SIM05680
	GAMMA(1,I)=(WN**(-2))*(1.0D0-EGT*COSW1T -EGT*(GMA/W1)*SINW1T)	SIM05690
	GAMMA(2,I) = (W1**(-1))*EGT*SINW1T	SIM05700
C		SIM05710
C		SIM05720
C		SIM05730
C		SIM05740
	ENDIF	SIM05750
C		SIM05760
C		SIM05770
C		SIM05780
600	CONTINUE	SIM05790
C		SIM05800
C		SIM05810
	R = 1	SIM05820
C		SIM05830
	DO 610 K = 1 ,100	SIM05840
C		SIM05850
C		SIM05860
C		SIM05870
C		SIM05880
C		SIM05890
	A(R,R) = PHI(1,1,K)	SIM05900
	A(R,R+1) = PHI(1,2,K)	SIM05910
	A(R+1,R) = PHI(2,1,K)	SIM05920
	A(R+1,R+1) = PHI(2,2,K)	SIM05930
C		SIM05940
C		SIM05950
C		SIM05960
C		SIM05970
C		SIM05980
C	*** B MATRIX FOR MULTIPLYING CONTROL TORQUES	SIM05990
C		SIM06000
	B(R,1) = GAMMA(1,K)*DBLE(UGVEX(412,K))	SIM06010
	B(R,2) = GAMMA(1,K)*DBLE(UGVEX(413,K))	SIM06020
	B(R,3) = GAMMA(1,K)*DBLE(UGVEX(414,K))	SIM06030
	B(R+1,1) = GAMMA(2,K)*DBLE(UGVEX(412,K))	SIM06040

	B(R+1,2) = GAMMA(2,K)*DBLE(UGVEX(413,K))	SIM06050
	B(R+1,3) = GAMMA(2,K)*DBLE(UGVEX(414,K))	SIM06060
C		SIM06070
C		SIM06080
C		SIM06090
C		SIM06100
C		SIM06110
C		SIM06120
C		SIM06130
C	*** BN MATRIX FOR MULTIPLYING THE NOISE DISTURBANCES	SIM06140
C		SIM06150
C		SIM06160
C		SIM06170
C		SIM06180
	BN(R,1)=GAMMA(1,K)*DBLE(UGVEX(ROWN1,K))	SIM06190
	BN(R,2)=GAMMA(1,K)*DBLE(UGVEX(ROWN2,K))	SIM06200
	BN(R,3)=GAMMA(1,K)*DBLE(UGVEX(ROWN3,K))	SIM06210
	BN(R+1,1)=GAMMA(2,K)*DBLE(UGVEX(ROWN1,K))	SIM06220
	BN(R+1,2)=GAMMA(2,K)*DBLE(UGVEX(ROWN2,K))	SIM06230
	BN(R+1,3)=GAMMA(2,K)*DBLE(UGVEX(ROWN3,K))	SIM06240
C		SIM06250
C		SIM06260
C		SIM06270
C		SIM06280
C		SIM06290
C		SIM06300
	R = R+2	SIM06310
610	CONTINUE	SIM06320
C		SIM06330
C	*** C MATRIX PRODUCTION ***	SIM06340
C		SIM06350
C		SIM06360
C		SIM06370
C		SIM06380
	JJ=-1	SIM06390
	DO 640 I=1,100	SIM06400
	JJ=JJ+1	SIM06410
	KK=I +JJ	SIM06420
C		SIM06430
C		SIM06440
C		SIM06450
	C(1,KK) = DBLE(UGVEX(418,I))	SIM06460
	C(2,KK) = DBLE(UGVEX(419,I))	SIM06470
	C(3,KK) = DBLE(UGVEX(420,I))	SIM06480
C		SIM06490
C		SIM06500
C		SIM06510
	KK=KK+1	SIM06520
C		SIM06530
	C(1,KK)=0.0	SIM06540
	C(2,KK)=0.0	SIM06550
	C(3,KK)=0.0	SIM06560
C		SIM06570
640	CONTINUE	SIM06580
C		SIM06590
C		SIM06600

C
C

RETURN
END

SIM06610
SIM06620
SIM06630
SIM06640

APPENDIX C. PROGRAM TO ESTIMATE NOISE IN KALMAN FILTER FROM UNOBSERVED MODES

```

C
C ***** SPA00010
C ***** SPA00020
C ***** SPAC 24 ***** SPA00030
C ***** ADAPTED TO RUN N MODES OF THE PLANT AND ***** SPA00040
C ***** COMPUTE THE NOISE IN THE KALMAN FILTER ***** SPA00050
C ***** FROM THE UNOBSERVED MODES ***** SPA00060
C ***** SPA00070
C ***** SPA00080
C ***** SPA00090
C ***** SPA00100
C ***** VARIABLE DECLARATIONS ***** SPA00110
C ***** SPA00120
C ***** SPA00130
EXTERNAL STMTRX,EXCMS SPA00140
CHARACTER*6 NAM SPA00150
CHARACTER*1 AGAIN,CORECT,RAGAIN SPA00160
INTEGER ROWN1,ROWN2,ROWN3,COUNT,NODE,MODE,KQ,EMODE,SMODE,R2M,C2M SPA00170
INTEGER CT,CF,KADJ,CFADJ,LOOP,PRNT,JJ,JK,N1,JR,KR,MR,ISEED,M2 SPA00180
INTEGER NO, NS,NF,SN, FN SPA00190
INTEGER JL,J1,JM , JP, JQ, KA, KB, KC, KD, KE, KF, KG SPA00200
C SPA00210
C SPA00220
C SPA00230
REAL LAMA(100), UGVEX(684,100),RNODE,RMODE,MIN SPA00240
REAL*8 PHI(2,2,100),GAMMA(2,100),EGT,GMA,WN,W1,X1T,X2T,TIME SPA00250
REAL*8 A(200,200),B(200,3),F(3, 50),IMPLSE,ENERGY SPA00260
REAL*8 COSW1T,SINW1T,X(200) SPA00270
REAL*8 DAMP,SAMPT,PI,SAMPTM,SUM1,SUM2,SUM3,SUMC SPA00280
REAL*8 C(9,200), RMN(3,3), QPN(3,3) SPA00290
REAL*8 BN(200,3) SPA00300
REAL*8 PNVARX, PNVARY, PNVARZ SPA00310
REAL*8 MNVARX, MNVARY, MNVARZ SPA00320
REAL*8 SUM, RNDM(6), RND1, RND2 SPA00330
REAL*8 ES,ED,ESUM,CGN,PRT SPA00340
REAL*8 WT , WD(3), BNWD(200), EX1(9) SPA00350
REAL*8 EX(9),AX(200) , SF , TO, CTT, ESS SPA00360
REAL*8 CTG, XDEL, XDEL1, ERS, PRT1 SPA00370
REAL*8 SF1 SPA00380
C SPA00390
C ***** SPA00400
C ***** VARIABLE DEFINITIONS ***** SPA00410
C ***** SPA00420
C ***** SPA00430
C STMTRX = SUBROUTINE EXTABLISHES STATE TRANSITION MATRICIES SPA00440
C LAMA = VECTOR OF THE SQUARE OF THE NATURAL FREQUENCIES SPA00450
C UGVEX = MODE POSITONS AND SLOPES OF THE NODAL POINTS SPA00460
C PHI = STATE TRANSITION MATRICIES FOR EACH MODE SPA00470
C GAMMA = INPUT TRANSITION MATRIX SPA00480
C A = DIAGONAL MATRIX CONSISTING OF PHI SPA00490

```

C	B = INPUT MATRIX OF GAMMA AND CONTROL SLOPES	SPA00500
C	DAMP = DAMPING FACTOR	SPA00510
C	SAMPT = SAMPLING TIME	SPA00520
C	IMPLSE = IMPULSE INPUT FUNCTION	SPA00530
C	MIN = NUMBER OF MINUTES SYSTEM WILL BE OBSERVED	SPA00540
C	SMODE = NUMBER OF STARTING MODE (INT)	SPA00550
C	MODE = NUMBER OF MODES (INT)	SPA00560
C	EMODE = NUMBER OF THE LAST MODE (INT)	SPA00570
C	NODE = NUMBER OF THE NOISE INPUT MODE (INT)	SPA00580
C	*** NOISE SLOPE LOCATIONS IN DATA MATRIX ***	SPA00590
C	ROWN1 = X-SLOPE LOCATION	SPA00600
C	ROWN2 = Y-SLOPE LOCATION	SPA00610
C	ROWN3 = Z-SLOPE LOCATION	SPA00620
C	C = OUTPUT MATRIX FOR Y	SPA00630
C	IDENT = IDENTITY MATRIX	SPA00640
C	RMN = MEASUREMENT NOISE COVARIANCE MATRIX	SPA00650
C	QPN = PLANT NOISE COVARIANCE MATRIX	SPA00660
C	PNVARX = PLANT NOISE X-SLOPE VARIANCE	SPA00670
C	PNVARY = PLANT NOISE Y-SLOPE VARIANCE	SPA00680
C	PNVARZ = PLANT NOISE Z-SLOPE VARIANCE	SPA00690
C	MNVARX = MEASUREMENT NOISE X-SLOPE VARIANCE	SPA00700
C	MNVARY = MEASUREMENT NOISE Y-SLOPE VARIANCE	SPA00710
C	MNVARZ = MEASUREMENT NOISE Z-SLOPE VARIANCE	SPA00720
C	ISEED = INITIALIZATION FOR RANDOM NUMBER GENERATOR	SPA00730
C	RNDM = RANDOM NUMBERS USED FOR WHITE NOISE IN MEASUREMENTS AND	SPA00740
C	IN PLANT FORCES	SPA00750
C	BN = B MATRIX TO MULTIPLY NOISE DISTURBANCES	SPA00760
C		SPA00770
C		SPA00780
C		SPA00790
C	***** SAMPLE OF SPACE EXEC FILE *****	SPA00800
C		SPA00810
C	THIS FILE MUST BEGIN IN COLUMN 1 AND RUN WITH THE FOLLOWING	SPA00820
C	SEQUENCE FOR THE INITIAL RUN OF THE PROGRAM:	SPA00830
C		SPA00840
C	FORTVS SPACE (COMPILES PROGRAM)	SPA00850
C	SPACE (EXECUTES EXEC FILE)	SPA00860
C	LOAD SPACE (START (LOADS AND EXECUTES PROGRAM)	SPA00870
C		SPA00880
C	SUBSEQUENT PROGRAM RUNS CAN ELIMINATE "FORTVS SPACE" IF NO	SPA00890
C	CHANGES HAVE BEEN MADE TO THE PROGRAM, AND CAN ELIMINATE	SPA00900
C	RUNNING THE EXEC FILE.	SPA00910
C		SPA00920
C	FI 4 DISK THESIS INPUT (PERM	SPA00930
C	FI 8 DISK UTILITY DATA (RECFM VS BLOCK 133 PERM	SPA00940
C	FI 11 DISK CNTRL OUTPUT (RECFM F BLOCK 80 LRECL 80 PERM	SPA00950
C	FI 13 DISK GAMMA OUTPUT (RECFM VS BLOCK 133 PERM	SPA00960
C	FI 14 DISK MODE OUTPUT (RECFM F BLOCK 80 LRECL 80 PERM	SPA00970
C	FI 16 DISK COST OUTPUT (RECFM F BLOCK 80 LRECL 80 PERM	SPA00980
C	FI 17 DISK PRT OUTPUT (RECFM F BLOCK 80 LRECL 80 PERM	SPA00990
C	FI 18 DISK ERROR DATA (RECFM F BLOCK 80 LRECL 80 PERM	SPA01000
C	FI 19 DISK END FILE (RECFM F BLOCK 80 LRECL 80 PERM	SPA01010
C	FI 20 DISK GMAT FILE (RECFM F BLOCK 80 LRECL 80 PERM	SPA01020
C		SPA01030
C	*****	SPA01040
C		SPA01050

	PARAMETER (JR=5243, KR=5397, MR=262139)	SPA01060
C		SPA01070
C		SPA01080
	MIN = 15.0	SPA01090
C		SPA01100
	WT=1.0D00	SPA01110
	PI = 4.0D0 * ATAN(1.0D0)	SPA01120
C		SPA01130
C	*****	SPA01140
C	***** READ LAMA AND UGVEX MATRICIES *****	SPA01150
C	*****	SPA01160
C		SPA01170
	CALL EXCMS ('CLRSCRN')	SPA01180
	WRITE(6,1008)	SPA01190
	WRITE(6,*) ' READING LAMA AND UGVEX MATRICIES'	SPA01200
	WRITE(6,*) ' ,	SPA01210
C	THIS SECTION READS THE LAMA VECTOR AND THE UGVEX	SPA01220
C	MATRIX AND STORES THEM IN MEMORY FOR FURTHER RECALL OF	SPA01230
C	DESIRED LOCATION DATA.	SPA01240
C		SPA01250
	READ(4,1001) NAM	SPA01260
	READ(4,1002)(LAMA(I),I=1,100)	SPA01270
	READ(4,1001) NAM	SPA01280
	DO 5 J = 1,100	SPA01290
	READ(4,1002)(UGVEX(I,J),I=1,684)	SPA01300
5	CONTINUE	SPA01310
C		SPA01320
1001	FORMAT(1X,A6)	SPA01330
1002	FORMAT(1X,8E15.8)	SPA01340
1008	FORMAT(1X,////)	SPA01350
C		SPA01360
500	CALL EXCMS ('CLRSCRN')	SPA01370
C		SPA01380
C	***** STARTING MODE NUMBER *****	SPA01390
C	** SMODE 7 TO 100 (INTEGER) ****	SPA01400
	SMODE= 17	SPA01410
C		SPA01420
	WRITE (16,700) SMODE	SPA01430
700	FORMAT (' ','STARTING MODE NUMBER: ',I2)	SPA01440
C		SPA01450
C	***** NUMBER OF MODES TO SCAN *****	SPA01460
C	** MODE 1 TO 93 (INTEGER)	SPA01470
C		SPA01480
	MODE=3	SPA01490
C		SPA01500
	EMODE = SMODE + MODE - 1	SPA01510
C		SPA01520
	WRITE (16,701) MODE	SPA01530
701	FORMAT (' ','NUMBER OF MODES SCANNED: ',I2)	SPA01540
C		SPA01550
C	***** NOISE INPUT POSITION *****	SPA01560
C	** NODE 1 TO 114 (INTEGER) (IF 0 THEN NO NOISE INPUT)	SPA01570
	NODE= 8	SPA01580
C		SPA01590
	WRITE (16,702) NODE	SPA01600
702	FORMAT (' ','NOISE NODE LOCATION: ',I5)	SPA01610

C		SPA01620
C	***** START AND STOP FOR PLANT	SPA01630
	SN=17	SPA01640
	FN=4	SPA01650
	NS=SN*2-1	SPA01660
	NF=SN*2+2*FN-2	SPA01670
	WRITE (16,899) SN, FN	SPA01680
899	FORMAT (' ', 'PLANT -- SN= ', I5, ' FN= ', I5)	SPA01690
C	***** SAMPLING TIME *****	SPA01700
C	** SAMPT MUST BE LESS THAN OR EQUAL TO SAMPTM **	SPA01710
	SAMPT = 0.05	SPA01720
	SAMPTM = ((2.0D0*PI)/SQRT(LAMA(EMODE)))/1.0D01	SPA01730
	IF (SAMPT.GE.SAMPTM) THEN	SPA01740
	SAMPT=SAMPTM	SPA01750
	ENDIF	SPA01760
C		SPA01770
	WRITE (16,900) MIN	SPA01780
900	FORMAT (' ', 2X, 'MIN: ', F8.3)	SPA01790
C		SPA01800
	WRITE (16,703) SAMPT, SAMPTM	SPA01810
703	FORMAT (' ', 'SAMPLING TIME: ', D12.4, 2X, 'SAMPTM= ', D15.8)	SPA01820
C		SPA01830
C	***** DAMPING FACTOR *****	SPA01840
C	** DAMP 0.0 TO 1.0 (REAL*8)	SPA01850
	DAMP=.01	SPA01860
C		SPA01870
	WRITE (16,704) DAMP	SPA01880
704	FORMAT (' ', 'DAMPING FACTOR: ', D12.4)	SPA01890
C		SPA01900
	NO=3	SPA01910
C	*** PLANT NOISE VARIANCE ***	SPA01920
C	** PNVARX, PNVAR, PNVARZ GT 0.0	SPA01930
C		SPA01940
	SF=1.0D0	SPA01950
	SF1=2.5D06	SPA01960
	PNVARX=1.0D00*Sf1	SPA01970
	PNVAR=1.0D00*Sf1	SPA01980
	PNVARZ=1.0D00*Sf1	SPA01990
C		SPA02000
C		SPA02010
C		SPA02020
C	*** MEASUREMENT NOISE VARIANCE ***	SPA02030
C	** MNVARX, MNVAR, MNVARZ GT 0.0	SPA02040
	MNVARX=1.0D-03 *SF	SPA02050
	MNVAR=1.0D-03 *SF	SPA02060
	MNVARZ=1.0D-03 *SF	SPA02070
C		SPA02080
C		SPA02090
	WRITE (16,711)	SPA02100
711	FORMAT(' ', 'PLANT NOISE VARIANCE: ')	SPA02110
	WRITE (16,712)	SPA02120
712	FORMAT(' ', 6X, 'PNVARX', 13X, 'PNVAR', 13X, 'PNVARZ')	SPA02130
	WRITE (16,713) PNVARX, PNVAR, PNVARZ	SPA02140
713	FORMAT(' ', 2X, E15.8, 2X, E15.8, 2X, E15.8)	SPA02150
	WRITE(16,714)	SPA02160
714	FORMAT(' ', 'MEASUREMENT NOISE: ')	SPA02170

	WRITE(16,715)	SPA02180
715	FORMAT(' ',6X,'MNVARX',13X,'MNVARY',13X,'MNVARZ')	SPA02190
	WRITE(16,713) MNVARX,MNVARY,MNVARZ	SPA02200
C		SPA02210
510	CALL EXCMS ('CLRSCRN')	SPA02220
	WRITE (6,1008)	SPA02230
	WRITE (6,*) ' PROGRAM RUNNING'	SPA02240
C		SPA02250
C	***** NOISE INPUT LOCATION *****	SPA02260
C		SPA02270
	ROWN3 = NODE*6	SPA02280
	ROWN2 = (NODE*6) - 1	SPA02290
	ROWN1 = (NODE*6) - 2	SPA02300
	COUNT = 0	SPA02310
C		SPA02320
C	***** INITIALIZE MATRICIES *****	SPA02330
C		SPA02340
	DO 54 K = 1, 200	SPA02350
	X(K) = 0.0	SPA02360
54	CONTINUE	SPA02370
C		SPA02380
	DO 60 I=1,3	SPA02390
	DO 58 J=1,3	SPA02400
	RMN(I,J)=0.0	SPA02410
	QPN(I,J)=0.0	SPA02420
58	CONTINUE	SPA02430
60	CONTINUE	SPA02440
C		SPA02450
	RMN(1,1)=MNVARX**2.0	SPA02460
	RMN(2,2)=MNVARY**2.0	SPA02470
	RMN(3,3)=MNVARZ**2.0	SPA02480
	QPN(1,1)=PNVARX**2.0	SPA02490
	QPN(2,2)=PNVARY**2.0	SPA02500
	QPN(3,3)=PNVARZ**2.0	SPA02510
C		SPA02520
C	*****	SPA02530
C	***** BEGIN MAIN PROGRAM *****	SPA02540
C	*****	SPA02550
C		SPA02560
	CALL STMTRX(EMODE,SMODE,SAMPT,DAMP,PHI,GAMMA,A,B,LAMA,UGVEX,C,	SPA02570
	+ ROWN1,ROWN2,ROWN3,BN)	SPA02580
C		SPA02590
C		SPA02600
	WRITE (6,1008)	SPA02610
	WRITE(6,*) ' EXIT STMTRX - - - PRE-LOOP KALMAN'	SPA02620
C		SPA02630
C		SPA02640
	WRITE (6,*) 'COMPUTING C TIMES SF FOR NEW C'	SPA02650
C		SPA02660
	WRITE (16,1008)	SPA02670
	DO 11000 I=1,200	SPA02680
	DO 10900 J=1,NO	SPA02690
	C(J,I)= C(J,I)*SF	SPA02700
10900	CONTINUE	SPA02710
11000	CONTINUE	SPA02720
C		SPA02730

C	*** PRE-LOOP PORTION OF KALMAN FILTER	SPA02740
C		SPA02750
C		SPA02760
	JK=SMODE*2-2	SPA02770
	M2=2*MODE	SPA02780
C		SPA02790
C		SPA02800
C	*****	SPA02810
C		SPA02820
	M2=2*MODE	SPA02830
	JP=2*SMODE-1	SPA02840
	JQ=2*EMODE	SPA02850
C		SPA02860
C		SPA02870
	DO 8813 I=1,3	SPA02880
	EX(I)=0.0	SPA02890
8813	CONTINUE	SPA02900
C		SPA02910
C		SPA02920
C		SPA02930
C		SPA02940
C	*****	SPA02950
C	***** THIS SECTION COMPUTES THE STATE UPDATE *****	SPA02960
C	*****	SPA02970
	ESS =0.0	SPA02980
	COUNT = 0	SPA02990
	ENERGY = 0.0D0	SPA03000
	TIME = 0.0	SPA03010
	CGN=0.0	SPA03020
	CTG=0.0	SPA03030
C	***** SETS LOOP FOR THE ITERATIONS NECESSARY TO OBSERVE *****	SPA03040
C	***** THE SYSTEM FOR THE NUMBER OF MINUTES SPECIFIED *****	SPA03050
	WRITE (6,1008)	SPA03060
	WRITE (6,*) ' START STATE UPDATE '	SPA03070
	LOOP = INT((MIN*60.0)/SAMPT)	SPA03080
	PRT= (DBLE(LOOP))/30.0	SPA03090
	PRT1=(DBLE(LOOP))/50.00	SPA03100
	CTT=0.0	SPA03110
C		SPA03120
	DO 400 L = 0, LOOP	SPA03130
	TIME = DBLE(L)*SAMPT	SPA03140
C		SPA03150
	IF(L.EQ.0)THEN	SPA03160
	IMPLSE =(1.0D06*SF1)/(DSQRT(SAMPT))	SPA03170
	ELSE	SPA03180
	IMPLSE = 0.0D0	SPA03190
	ENDIF	SPA03200
C		SPA03210
	TO=0.0	SPA03220
C	***** RANDOM NUMBER GENERATOR *****	SPA03230
C		SPA03240
	DO 101 I=1,6	SPA03250
	ISEED=MOD(ISEED*JR+KR,MR)	SPA03260
	RND1=(DBLE(ISEED)+0.5D00)/DBLE(MR)	SPA03270
	ISEED=MOD(ISEED*JR+KR,MR)	SPA03280
	RND2=(DBLE(ISEED)+0.5D00)/DBLE(MR)	SPA03290

	RNDM(I)=DSQRT(-2.0*DLOG(RND1))*DCOS(6.2831853D00*RND2)	SPA03300
101	CONTINUE	SPA03310
C	*****	SPA03320
	CTT=CTT+1.0	SPA03330
C	***** START OF STATE UPDATE *****	SPA03340
C		SPA03350
C	**** COMPUTE AX ⁰ 200 = A ⁰ 200 X 200 * X ⁰ 200	SPA03360
C		SPA03370
C		SPA03380
C	**** COMPUTE AXH = A*XH	SPA03390
C		SPA03400
	JK=SMODE*2-2	SPA03410
	JP=JK+1	SPA03420
	JQ=2*EMODE	SPA03430
C		SPA03440
C		SPA03450
	DO 5015 I=NS,NF	SPA03460
	SUM=0.0	SPA03470
	DO 5010 K=NS,NF	SPA03480
	SUM=SUM+A(I,K)*X(K)	SPA03490
5010	CONTINUE	SPA03500
	AX(I)=SUM	SPA03510
5015	CONTINUE	SPA03520
C		SPA03530
C		SPA03540
C	**** COMPUTE WD ⁰ 3	SPA03550
C		SPA03560
	WD(1)=PNVARX*RNDM(1)*TO+IMPLSE	SPA03570
	WD(2)=PNVARY*RNDM(2)*TO	SPA03580
	WD(3)=PNVARZ*RNDM(3)*TO	SPA03590
C		SPA03600
C	**** COMPUTE BNWD ⁰ 200 =BN ⁰ 200 X 3 * WD ⁰ 3	SPA03610
C		SPA03620
	DO 5035 I=NS,NF	SPA03630
	SUM=0.0	SPA03640
	DO 5030 K=1,3	SPA03650
	SUM=SUM+BN(I,K)*WD(K)	SPA03660
5030	CONTINUE	SPA03670
	BNWD(I)=SUM	SPA03680
5035	CONTINUE	SPA03690
C		SPA03700
C	**** COMPUTE X ⁰ 200 =AX ⁰ 200 + BNWD ⁰ 200	SPA03710
C		SPA03720
	DO 5040 I=NS,NF	SPA03730
	X(I)= AX(I) + BNWD(I)	SPA03740
	IF (DABS(X(I)).LT. 1.0D-60) THEN	SPA03750
	X(I)=1.0D-60	SPA03760
	END IF	SPA03770
C		SPA03780
5040	CONTINUE	SPA03790
C		SPA03800
C		SPA03810
C	*****	SPA03820
C		SPA03830
C	**** START OF KALMAN FILTER ****	SPA03840
C		SPA03850

	JK=SMODE*2-2	SPA03860
	JP=JK+1	SPA03870
	JQ=2*EMODE	SPA03880
	M2=2*MODE	SPA03890
C		SPA03900
	JL=JQ+1	SPA03910
	DO 8888 I=1,NO	SPA03920
	SUM=0.0	SPA03930
	DO 8887 K=JL,NF	SPA03940
	SUM=SUM+C(I,K)*X(K)	SPA03950
8887	CONTINUE	SPA03960
	EX(I)=SUM*SUM*SAMPT+EX(I)	SPA03970
8888	CONTINUE	SPA03980
C		SPA03990
C		SPA04000
	CGN=CGN+1.0	SPA04010
	IF (CTT.EQ.1.0.OR.CGN.GT.PRT) THEN	SPA04020
C		SPA04030
	WRITE (16,1008)	SPA04040
	WRITE (16,*) 'TIME = ', TIME	SPA04050
C		SPA04060
	DO 380 I=JP , JQ	SPA04070
	WRITE (16,4500) I,X(I)	SPA04080
380	CONTINUE	SPA04090
4500	FORMAT (' ',2X,'X(',I4,')= ',D15.8)	SPA04100
C		SPA04110
	CGN=0.0	SPA04120
	END IF	SPA04130
C		SPA04140
C		SPA04150
400	CONTINUE	SPA04160
	WRITE (11,*) 'SMODE = ', SMODE	SPA04170
	WRITE (11,*) 'EMODE = ', EMODE	SPA04180
	WRITE (11,*) 'SN = ',SN	SPA04190
	WRITE (11,*) 'FN = ',FN	SPA04200
C		SPA04210
	JL=JQ+1	SPA04220
	DO 9499 I=1,NO	SPA04230
	WRITE (11,*) 'EX ',I , ' ', EX(I)	SPA04240
9499	CONTINUE	SPA04250
C		SPA04260
C		SPA04270
C		SPA04280
C		SPA04290
C		SPA04300
C		SPA04310
	CALL EXCMS ('CLRSCRN')	SPA04320
	WRITE (6,1008)	SPA04330
C		SPA04340
599	STOP	SPA04350
	END	SPA04360
C		SPA04370
C		SPA04380
C		SPA04390
C		SPA04400
C		SPA04410

C	*****	SPA04420
C	THIS SUBROUTINE COMPUTES THE STATE TRANSITION MATRIX FOR EACH	SPA04430
C	OF THE 100 MODES	SPA04440
C	*****	SPA04450
C		SPA04460
	SUBROUTINE STMTRX(EMODE,SMODE,T,D,PHI,GAMMA,A,B,LAMA,UGVEX,C,	SPA04470
+	+ ROWN1,ROWN2,ROWN3,BN)	SPA04480
C		SPA04490
	REAL*8 WN,GMA,PHI(2,2,100),GAMMA(2,100),EGT,T,COSW1T,SINW1T	SPA04500
	REAL*8 W1,D,A(200,200),B(200,3),C(9,200),BN(200,3)	SPA04510
	REAL LAMA(100),UGVEX(684,100)	SPA04520
	INTEGER SMODE,R,EMODE,JJ,KK,ROWN1,ROWN2,ROWN3, NN(9), N9, NO	SPA04530
C		SPA04540
C		SPA04550
	WRITE (6,*) 'INSIDE STMTRX - - COMPUTE WN, GMA, EFT, W1'	SPA04560
C		SPA04570
C		SPA04580
	DO 600 I = 1, 100	SPA04590
	WN = DBLE(SQRT(LAMA(I)))	SPA04600
	GMA = D*WN/2.0	SPA04610
	EGT = DEXP(-GMA*T)	SPA04620
	W1 = DSQRT((WN**2)-(GMA**2))	SPA04630
	COSW1T = DCOS(W1*T)	SPA04640
	SINW1T = DSIN(W1*T)	SPA04650
C		SPA04660
C		SPA04670
C		SPA04680
C		SPA04690
	IF(WN.EQ.0)THEN	SPA04700
	PHI(1,1,I) = EGT*COSW1T	SPA04710
	PHI(1,2,I) = T	SPA04720
	PHI(2,1,I) = 0	SPA04730
	PHI(2,2,I) = EGT*COSW1T	SPA04740
C		SPA04750
C		SPA04760
C		SPA04770
C		SPA04780
C		SPA04790
C		SPA04800
	GAMMA(1,I) = 0	SPA04810
	GAMMA(2,I) = 0	SPA04820
	ELSE	SPA04830
C		SPA04840
C		SPA04850
C		SPA04860
C		SPA04870
	PHI(1,1,I) = EGT*(COSW1T + (GMA*(W1**(-1)))*SINW1T)	SPA04880
	PHI(1,2,I) = (W1**(-1))*EGT*SINW1T	SPA04890
	PHI(2,1,I) = -(WN**2)*(W1**(-1))*EGT*SINW1T	SPA04900
	PHI(2,2,I) = EGT*(COSW1T - (GMA*(W1**(-1)))*SINW1T)	SPA04910
C		SPA04920
C		SPA04930
C		SPA04940
C		SPA04950
	GAMMA(1,I)=(WN**(-2))*(1.0D0-EGT*COSW1T -EGT*(GMA/W1)*SINW1T)	SPA04960

	GAMMA(2,I) = (W1**(-1))*EGT*SINW1T	SPA04970
C		SPA04980
C		SPA04990
C		SPA05000
C	ENDIF	SPA05010
C		SPA05020
C		SPA05030
C		SPA05040
C		SPA05050
600	CONTINUE	SPA05060
C		SPA05070
	WRITE (6,*) 'PHI AND GAMMA COMPUTED'	SPA05080
	WRITE (6,*) ' COMPUTING A, B, BN'	SPA05090
C		SPA05100
	R = 1	SPA05110
C		SPA05120
	DO 610 K = 1 ,100	SPA05130
C		SPA05140
C		SPA05150
C		SPA05160
C		SPA05170
C		SPA05180
	A(R,R) = PHI(1,1,K)	SPA05190
	A(R,R+1) = PHI(1,2,K)	SPA05200
	A(R+1,R) = PHI(2,1,K)	SPA05210
	A(R+1,R+1) = PHI(2,2,K)	SPA05220
C		SPA05230
C		SPA05240
C		SPA05250
C		SPA05260
C		SPA05270
C	*** B MATRIX FOR MULTIPLYING CONTROL TORQUES	SPA05280
C		SPA05290
	B(R,1) = GAMMA(1,K)*DBLE(UGVEX(412,K))	SPA05300
	B(R,2) = GAMMA(1,K)*DBLE(UGVEX(413,K))	SPA05310
	B(R,3) = GAMMA(1,K)*DBLE(UGVEX(414,K))	SPA05320
	B(R+1,1) = GAMMA(2,K)*DBLE(UGVEX(412,K))	SPA05330
	B(R+1,2) = GAMMA(2,K)*DBLE(UGVEX(413,K))	SPA05340
	B(R+1,3) = GAMMA(2,K)*DBLE(UGVEX(414,K))	SPA05350
C		SPA05360
C		SPA05370
C		SPA05380
C		SPA05390
C		SPA05400
C		SPA05410
C		SPA05420
C	*** BN MATRIX FOR MULTIPLYING THE NOISE DISTURBANCES	SPA05430
C		SPA05440
C		SPA05450
C		SPA05460
C		SPA05470
	BN(R,1)=GAMMA(1,K)*DBLE(UGVEX(ROWN1,K))	SPA05480
	BN(R,2)=GAMMA(1,K)*DBLE(UGVEX(ROWN2,K))	SPA05490
	BN(R,3)=GAMMA(1,K)*DBLE(UGVEX(ROWN3,K))	SPA05500
	BN(R+1,1)=GAMMA(2,K)*DBLE(UGVEX(ROWN1,K))	SPA05510
	BN(R+1,2)=GAMMA(2,K)*DBLE(UGVEX(ROWN2,K))	SPA05520

	BN(R+1,3)=GAMMA(2,K)*DBLE(UGVEX(ROWN3,K))	SPA05530
C		SPA05540
C		SPA05550
C		SPA05560
C		SPA05570
C		SPA05580
C		SPA05590
	R = R+2	SPA05600
610	CONTINUE	SPA05610
C		SPA05620
	WRITE (6,*) 'A, B, BN COMPUTED'	SPA05630
C		SPA05640
	WRITE (6,*) 'COMPUTING C'	SPA05650
C		SPA05660
C		SPA05670
C		SPA05680
C	*** C MATRIX PRODUCTION ***	SPA05690
	NO=3	SPA05700
	NN(1)=418	SPA05710
	NN(2)=419	SPA05720
	NN(3)=420	SPA05730
C		SPA05740
C		SPA05750
C		SPA05760
C		SPA05770
C		SPA05780
C		SPA05790
	JJ=-1	SPA05800
	DO 640 I=1,100	SPA05810
	JJ=JJ+1	SPA05820
C		SPA05830
	DO 9127 K=1,NO	SPA05840
C		SPA05850
	KK=I+JJ	SPA05860
C		SPA05870
	N9=NN(K)	SPA05880
C		SPA05890
	C(K,KK) = DBLE(UGVEX(N9,I))	SPA05900
C		SPA05910
	KK=KK+1	SPA05920
C		SPA05930
	C(K,KK)=0.0	SPA05940
9127	CONTINUE	SPA05950
640	CONTINUE	SPA05960
C		SPA05970
C		SPA05980
C		SPA05990
	RETURN	SPA06000
	END	SPA06010

LIST OF REFERENCES

1. Rodriguez, G., *Control Technology Development*, Large Space Systems Technology - 1980, NASA Conference Publication 2168, Vol. I, pp. 49-63, November 18-20, 1980.
2. Thomson, William T., *Theory of Vibration with Applications*, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1988.
3. Preston, William J. , *Effects of Reduced Order Modeling on the Control of a Large Space Structure*, Masters Thesis, Naval Postgraduate School, CA, September 1988.
4. Selby, Samuel M., *Standard Mathematical Tables*, The Chemical Rubber Co., Cleveland, OH, 1972.
5. Mendel, Jerry M., *Discrete Techniques of Parameter Estimation*, Marcel Dekker, Inc., NY, 1973.
6. Nishimura, T., "On the a priori Information in Sequential Estimation Problems," *Kalman Filtering: Theory and Application*, p. 138-145, Edited by: Sorenson, H. W., IEEE Press, Inc., New York, 1985.
7. Franklin, Gene F. and Powell, David J., *Digital Control of Dynamic Systems*, Addison-Wesley Company, Reading, MA, 1980.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, VA 22304-6145	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, CA 93943-5002	2
3.	Director Naval Research Laboratory Washington, D.C. 20375	1
4.	Commander Naval Space Command Attn: Code N3 Dahlgren, VA 22448-5170	1
5.	Chairman, Code 62 Department of Electrical and Computer Engineering Naval Postgraduate School Monterey, CA 93943	1
6.	Prof. Jeff B. Burl, Code 62BL Department of Electrical and Computer Engineering Naval Postgraduate School Monterey, CA 93943	1
7.	Prof. H. A. Titus, Code 62TS Department of Electrical and Computer Engineering Naval Postgraduate School Monterey, CA 93943	1
8.	Major W. J. Preston 7 Cary Place Fredricksburg, VA 22405	1
9.	Lcdr. Bruce M. Jackson c/o Col. Jose A. Aybar (USA-Ret) 22577 Veronica Drive Salinas, CA 93943	2

Thesis
J2125
c.1

Jackson

Utilization of a Kal-
man observer with large
space structures.

15 APR 92

37043

Thesis

J2125 Jackson

c.1 Utilization of a Kal-
man observer with large
space structures.



thesJ2125
Utilization of a Kalman observer with la



3 2768 000 81616 9
DUDLEY KNOX LIBRARY